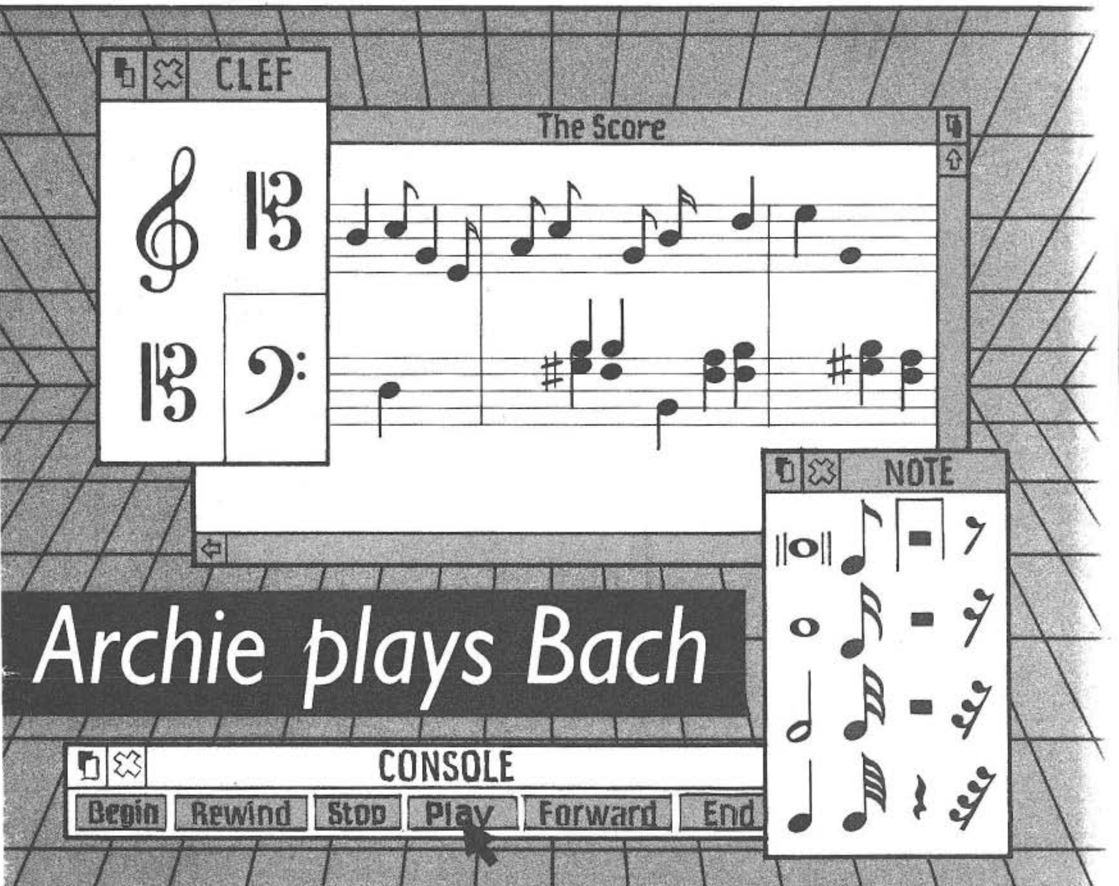
Volume 1
Issue 2

December
1987

Price £1.20

RISC USER



Archie plays Bach

THE MAGAZINE AND SUPPORT GROUP
EXCLUSIVELY FOR USERS OF THE ARCHIMEDES

Volume 1 Issue 2
December 1987

RISC USER

CONTENTS

RISC User is published by BEEBUG Ltd.
Co-Editors: Mike Williams, Lee Calcraft
Assistant Editor: Kristina Lucas
Production Assistant: Yolanda Turuelo
Subscriptions: Mandy Mileham

BEEBUG, Dolphin Place, Holywell Hill, St.Albans,
Herts AL1 1EX. Tel. St.Albans (0727) 40303

The Publisher cannot accept any responsibility whatsoever for errors in articles, programs, or advertisements published. The opinions expressed on the pages of this journal are those of the authors and do not necessarily represent those of the Publisher, BEEBUG Limited.

The Archimedes Magazine
and Support Group.

EDITORIAL

By the time that you read these columns you will probably be aware that Acorn have recently made some 50 staff redundant. They assure us however, that their support for the Archimedes will not be impaired, and that they will be concentrating more, rather than less, on this machine and its successors. The announcement of these redundancies comes just weeks after Acorn reduced the price of the Archimedes range by £100, and within a few days of announcements that their language software for the machine, including C, Pascal and Fortran is to be increased in price from around £100 to around £180.

BBC micro users are used to a degree of uncertainty by the marketing managers at Acorn. The original BBC micro was increased in price just a short while after its launch, and a few months ago the price of the Master Compact first dropped by £100, and then went up again by £50, all within the space of a fortnight. In our view it is important for Acorn to make firm marketing decisions, and then stick to them. Price fluctuations within a few months of the launch of products such as C or the Archimedes itself do little to inspire confidence, and at worst positively alienate the customers on whom Acorn relies.

We are of course happy that the Archimedes is now less expensive, and therefore available to a wider public, but we feel less than happy for those who showed their confidence in Acorn, and bought their Archimedes too early. As for the language price rise, we would strongly urge Acorn to reconsider. They might take as their model the hugely successful Borland, whose C and other languages for the IBM PC have become highly respected market leaders - and all for under £100 per copy. A product does not have to cost £200 to prove that it is any good, as Borland have demonstrated to the detriment of those who continue to market IBM language software at the £200 or £300 mark.

The response to the first issue of RISC User has been very encouraging, and we would urge readers to continue to write in, either with ideas for subjects that you would like to see covered, or with contributions to the magazine. We welcome all types of articles, programs, hints and tips etc.

As readers who attended the Micro User show will know, we are including on each magazine disc, whenever possible, extra items of interest. The first disc contains all the programs from issue one of the magazine, and runs from a WIMPs menu. The disc also contains a stunning 3D animation of Newton's Cradle written by Computer Concepts. This really shows off the power and speed of the Archimedes, though the code is 600K long, and you will need a 310 to make it work. Issue two of the disc will contain a set of graphics screens created by Clares with their Artisan package, plus a couple of demo screens from their Image Writer, reviewed in this issue. The music file for Bach's Minuet in G is also included this month.

NewsNewsNewsNewsNews

A round-up of the latest news and comment in the Archimedes world compiled by Mike Williams. All prices quoted below include VAT.

ACORN

Acornsoft Prolog and Lisp are now available at the new higher price levels of £228.85 (£217.41 to RISC User members). It is expected that this price will in future be applied to all Acornsoft languages for the Archimedes, so buy 'C', Fortran or Pascal soon and you should be able to find copies at the old price of £118.85 (£108.16 to members).

A significant boost to Archimedes sales has been achieved by Acorn's 0% finance facilities. This allows the balance of the purchase price to be paid off over 12 months at no extra cost. The scheme is available through BEEBUG and most other Acorn dealers.

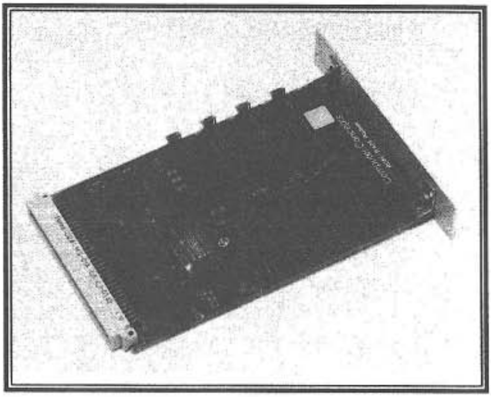
Latest news on applications software is that Logistix (the PC spreadsheet and planning import) has hit a bug. Although this is also present on PC versions, the Acorn version fails to trap this and leaves the user with a 'C' language error message unlike the PC version where the error is trapped before it can affect the user! Although Acorn started to send out the replacement Series One operating system in early December to all registered Archimedes users, the freebie word processor ArcWriter was left out. This will now be sent separately (rumour has it when Acorn get your old OS chips back!).

BEEBUG

Three items for Archimedes users are available now from BEEBUG. For those upgrading from older BBC micros, a buffered interface for connecting 5.25" disc drives to the Archimedes costs £24.00 and allows up to four external drives to be accessed for file transfer. We have had a prototype working with two drives for some while now. A Serial Link kit is available for connecting a BBC model B or Master Series computer to an Archimedes using the serial ports. The kit includes connecting lead and software on disc for £17.25. Adding to the applications available for the Archimedes, BEEBUG has released a version of its successful Masterfile II database package. This runs under Basic V (not just the Emulator), thus offering worthwhile increases in speed, and now provides 80 column screens for appropriate functions. Masterfile II for the Archimedes costs just £16.50 to members (£22.00 to non-members).

The digital to analogue monitor conversion kit is taking longer to finalise than expected, and will not now be ready until February 1988 (price still to be fixed). The BEEBUG modem podule and communications software are progressing well, and first deliveries are expected for late January 1988. Full details of BEEBUG products for the Archimedes will be included with every RISC User mailing.

COMPUTER CONCEPTS



The Computer Concepts ROM podule should be available by the time you read this. It works very much like a BBC micro ROM board, and will take 7 ROMs, EPROMs or RAM chips, each of which may be up to 128K in size. It has optional battery backup, and can provide the user with a total of 896K of extra storage space. The board fits inside an Archimedes plugging into the podule backplane, an optional add-on at £44.85 (£42.61 to members). The backplane provides a total of two sockets for add-ons, so there is still room for one further board after CC's has been fitted. Computer Concepts are on (0442) 639933 and the podule costs £56.35 (£53.53 to members).

WILD VISION

Chromalock from improbably named Wild Vision is a low cost genlock (for synchronising the Beeb's video signal for input to a VCR), and full colour video overlay system which allows the spectacular graphics of the Archimedes computer to be superimposed onto a video picture. Chromalock has scientific, industrial and educational applications, and its state-of-the-art circuitry also produces high-quality results with most domestic VCRs. Prices range from £155 for a genlock only system to £355 for a professional de luxe system. Wild Vision are on 091-281 8481.

SYNCROTECH

Syncrotech, formerly Expert Software, has announced a low cost communications package for the Archimedes. This is claimed to work with Demon and Designer modems, and also the WS4000, Linnet, Nightingale and Magic Modems. The final price, still to be fixed, is expected to be just under £25, and the product should be ready this December. For more information contact Syncrotech on 01-959 7535. RU

THE ARCHIMEDES SERIES ONE UPGRADE

Acorn is mailing the series one upgrade to all registered users during December. Lee Calcraft has compiled some notes on what you get.

First of all, you must have sent back your registration card in order to qualify for the free upgrade. The pack will consist of a new set of ROMs, plus a new Welcome disc and Welcome Guide. And if you are one of those who received the very early 0.2 operating system (type *FX0 to find out), then you will also get a new Basic User Guide. The free ArcWriter word processor will follow in a few weeks.

The Welcome disc contains a few enhancements. For example, you will get a new version of the Music Editor (see "Archie Plays Bach" in this issue), and two music files will also be supplied, so you can actually play something now! But the more important changes result from the new operating system.

THE NEW OPERATING SYSTEM

When you turn on your machine after the upgrade, you will find that you are in the Desktop. What Acorn have done is to put the massive Desktop program into ROM as a relocatable module - even though it is still in Basic. Some commentators regard this as a waste of valuable ROM space. The Desktop takes up over 98K - nearly twice as much as the ARM Basic interpreter. Moreover, you cannot even use the Desktop from within an application such as Basic or a word processor to display and alter directories (contrast this with the 9K RISC User Disc Menu).

Of course, it is a neat trick to start up in the Desktop, and this may well impress would-be purchasers; but if you prefer to start up in Basic, quit the Desktop, and reconfigure your machine using:

```
*CONFIGURE LANGUAGE 4
```

then press Ctrl-Break. Altering the value to 3 will revert to the Desktop. You can in any case call the Desktop at any time with *DESKTOP.

If you are writing software which will need to auto-boot on an Archimedes with the Desktop configured as the start-up language, then you must create a *runnable* !BOOT file (rather than an EXECable one). For example, it could be a short Basic program. You should then type:

```
*OPT 4,2  (not 4,3)
```

This will will now work whether the disc is booted from Basic or from within the Desktop.

MORE MODULES - LESS RAM

If you type *MODULES on the new system, you will find that you have 20 resident relocatable modules, including a number of items - such as the Debugger, Basic Editor, Sound Voices, Printer Dumps (type "*HELP HARDCOPY.") - which were formerly on the Welcome disc. This is all very good news, especially since this saves RAM space, as each module runs directly from ROM. Unfortunately however, these modules still take up user RAM for workspace. You can get some idea of how much by subtracting &1800000 from the bottom non-zero address in the 'Workspace' column of the *MODULES display. It is around 110K! You may well wish to use `*UNPLUG modulename` to remove any which you are not using (especially if you have a 305). You can reinstate them with `*RMREINIT modulename`.

OTHER FEATURES

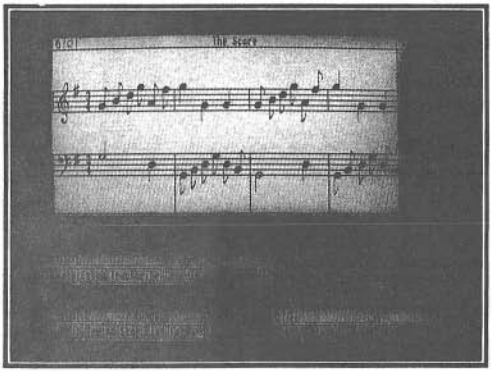
There are many other features to the new firmware. The annoying buzz generated on pressing Escape has been fixed, and Basic has been enhanced with a much more comprehensive HELP system. Type 'HELP' for further details. The *WIPE and *COPY options have been changed (see this month's Hints). Annoyingly, the *COPY option now takes you out of Basic and leaves you in Arthur if you perform a 'Quick' copy. This means that the RISC User Disc Menu cannot be used to copy multiple-tagged files in the 'Quick' mode. Maybe this will be rectified on later versions of the operating system? It is not clear whether Acorn will be producing further major releases of the operating system. But they are thought to be writing a new UNIX operating system for the Archimedes with true multi-tasking, which should be available by autumn 1988. RU

ARCHIE PLAYS BACH

Chris Hilderbrands shows how to use the Music Editor to enter the score of Bach's Minuet in G.

The Music Editor supplied on the Welcome disc is a nicely written piece of software which allows music to be transcribed from a score, and played at the touch of a key. The version supplied with machines using 0.2 or 0.3 operating systems has a number of limitations, but these have mostly been overcome in the full release version. This will be distributed on the Welcome disc which accompanies the series one upgrade, and will be mailed to all users in December.

This article refers to the series one version of the Music Editor, though where major differences occur, these will be noted in the text. Since the object of the article is to show how easy it is to enter music from a score, you will really need to have the program running for it to make any sense.



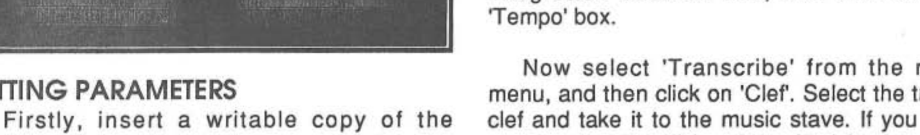
SETTING PARAMETERS

Firstly, insert a writable copy of the Welcome disc, and call the Desktop. Select the disc icon, then the 'Apps' directory, and finally the MusicEd file. Once the program has loaded, you will see a display of an empty score, and the 'Maestro' main menu. This menu can be recalled at any time by pressing the middle or *menu* button of the mouse.

Now select 'Help' (i.e. with the *select* or left button of the mouse), then select 'Describe'. This will place an active 'Help' box at the foot of the screen. Now click on the 'X' of the 'Help' box to remove the 'Help', but not the 'Describe' box. Next click on the 'Setup' box, and then select 'Stave'. The top three options here are 'Voice', 'Keyboard' and 'Chorus'. These provide for a single stave, two staves, or four staves respectively. For the purposes of Bach's Minuet in G, click on 'Chorus'. You can also select one or two lines of percussion from this menu, but we do not need them here.

Next, get rid of the 'Stave' box, and call up the 'Instrument' options from the 'Set up' menu. If you have used the early version of Maestro, you will be pleasantly surprised by this selection box, because as well as selecting instruments, you can also set up the volume of each, and the stereo position! For the Bach piece, select 'StringLib-Soft' on the top line of staves one and two. The defaults of 'Fortississimo' (sic), and stereo 'Centre' are both acceptable. You can alter any of these items as a piece is playing and hear the effects instantaneously. Now kill the 'Instrument' box, and select 'Tempo'. Move the pointer to 'Alegretto'. Click on this, and then kill the 'Tempo' box.

Now select 'Transcribe' from the main menu, and then click on 'Clef'. Select the treble clef and take it to the music stave. If you now click again, it will be deposited in the right place, without the problems experienced with earlier versions of the program. Next, select the bass clef, and put that in place. Here you may need to scroll the stave a little, or extend the



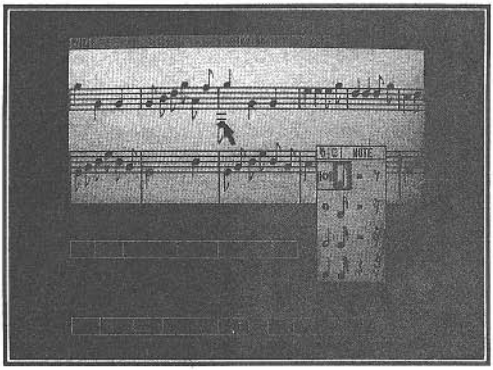
'Score' area downwards to bring the lower stave into view. When the clefs are in place, kill the 'Clef' box, and select 'Key'. Press the *adjust* button (far right) to alter the '0' to a '1', then select the '#' and move it to the top stave near to the treble clef. Click it in place, and kill the 'Key' box. Now select 'Time', and use the *adjust* button to obtain a '3' in the upper box. If you now select this and click it on the upper stave, you are finally ready to begin entering notes.

ENTERING NOTES

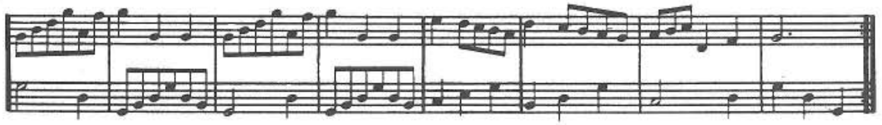
From the 'Transcribe' menu, select 'Note', and then drag the 'Note' box to somewhere convenient on the screen. All you need to do now is to move the pointer to the desired note (a green box will appear around the one that you select), then move the pointer to the stave, and position the note before pressing select to fix it. Any note can be deleted by placing an identical note exactly on top of it. In transcribing from a piece of sheet music, the idea is to make as close a copy as possible of the original score on the screen. The accompanying picture, shows our attempt.

To avoid timing problems when entering music from a score, it is a good idea to enter notes on the two staves together, bar by bar, doing whichever stave has the most notes in it first, and inserting the bar lines before moving to the second stave. Bar lines are inserted by selecting the 'Other' box on the 'Transcribe' menu. So, for the example piece, begin with the top stave 1st bar, followed by bottom stave 1st bar, then bottom 2nd and top 2nd etc. While you are entering notes, the 'Console' can be used to scroll the music along, or to play it at any time. This allows easy experimentation with the fruits of your labours.

When you are happy with the piece, select 'Repertoire' from the main menu. Then select 'Catalogue'. This new addition to the program will display a catalogue of the UserTunes directory of your disc. To save your score with all current settings, move the pointer to the line below the word 'Catalogue', and press *select*. Then type in a filename for the save operation, and press Return. Make sure that this remains highlighted in green, then select 'Save' from the 'Repertoire' menu, and your file will be saved. To load it back in, highlight the required filename by moving the pointer to it in the 'Catalogue' box. Then select 'Load' from the 'Repertoire' menu.



The results of the Bach piece are, with strings soft or plucked, fairly pleasing, if a little expressionless. If you try and use all 8 voices as in the example shown below, you can get a degree of expression, and a quite powerful sound. My main complaint about the revised version is that there still seems to be no *copy* or *coda* for the repetitive bits, (especially rhythms where you seem to need rests to be certain of the timing). Its other major limitation is that there is no voice generation routine. But, hopefully, before long, a voice generator will be available, and the sounds of orchestras, rather than massed harpsichords, will be heard across the land. RU



Archimedes Multi-Tone Printer Dump

by Lee Calcraft

Put all your smart Archimedes graphics designs onto paper using this general purpose screen dump for Epson compatible printers.

On operating system series one (to be distributed free of charge by Acorn in December) there are three (working) resident printer dumps. Although extremely useful, these are however, only monochrome dumps. To attempt to reproduce a coloured Archimedes screen, you will need a multi-tone dump; and this is where the accompanying program comes in. It can reproduce up to 16 different shades on an Epson compatible printer from any Archimedes graphics mode. This allows it to follow to some extent the Archimedes' range of grey scales. Moreover, the area of the screen dumped to the printer is defined using the mouse. This is particularly useful from a speed point of view, since although the full dump takes 4 minutes on an Epson FX 80, dumping a small part of the screen to check the output etc, is accomplished in a matter of seconds.

RUNNING THE PROGRAM

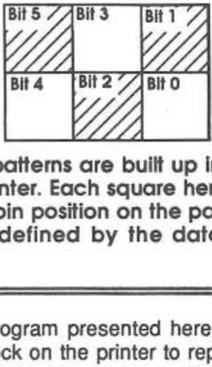
First of all, type in the program and save it away. When it is run, it will clear the screen, display the catalogue of the current disc directory, and ask for a screen filename. The screen is then loaded in from disc using the Archimedes *SCREENLOAD command. This, as you may know, is a little slow. For testing purposes you may like to skip this part of the program, and draw something on the screen to test the dump. To do this, replace lines 130 and 140 with your own text printing or drawing routine.

Once a screen has been drawn or loaded from disc, the pointer will appear, and you should then use the mouse to define the rectangle to be dumped. Click the *select* button (far left button) once to mark the top left corner of the rectangle, and drag the box out to mark the bottom right corner, again by pressing the *select* button. It is important to mark the corners in this order, otherwise the program will 'beep', and you will need to repeat the process. Once the area is correctly marked, the dump will proceed. On completion, a form feed will be issued, and the printer will be reset for normal text output.

If you wish to terminate the printout at any point, just press the Escape key, and leave it held down until the word *Escape* appears on the screen. The dumping process should terminate cleanly at the end of the next line, and a form feed will again be issued.

SHADES OF GREY

When writing computer printer dumps, considerable compromise is always called for. And the greater the computer's graphics resolution and its number of colours, the greater the compromise required. The problem is essentially how to represent a single graphics pixel from the screen, which may be any of 256 different logical colours, by a single printer pixel which can be either on or off. Obviously one way to represent the colours is to create shading on the printer using patterns of dots to represent a single pixel of any one colour. But if you do this, you would need a paper size of several feet square. Hence the need for compromise!



Shading patterns are built up in 3x2 blocks on the printer. Each square here represents a printer pin position on the paper. This 3x2 block is defined by the data sequence 100110.

In the program presented here, I have used a 3x2 pixel block on the printer to represent a pair of horizontally adjacent screen pixels in 640x256 pixel modes (modes 0, 8, 12 and 15). The routine then checks to see if the two adjacent pixels are the same. If so, it creates the appropriate pattern. If not, it splits the pattern between the two types of shading required for the two different pixels. The upshot of this is that we can use shading for any given colour based on a 3x2 printer dot pattern even in the Archimedes' higher resolution modes. This in turn allows us to create shades which can give some representation of the colours available in the sixteen colour modes.

The sixteen printer dot patterns are held in binary form in DATA statements at line 520. The first is used for colour zero (as returned by the POINT function), the second for colour one, and so on. The accompanying figure shows the correspondence between the binary data and the dot pattern which it represents. It is up to the user to adjust the DATA statements to suit his own particular application. For example, the data for colour zero has been set to zero, and a special routine (at line 450) ensures that the logical colour normally associated with white in any given mode is printed in black. This is ideal for normal text, but when printing certain screens, you may wish to reverse the effect. This is achieved by by swapping the 111111 in line 510 with the 000000 in line 520. All the intermediate shades may also need to be adjusted to suit any given picture. The reason why we cannot settle on a given set of shades to represent a given set of colours is that the designer of the screen picture has 4096 physical colours to play with, and he may set any given logical colour number to any one of these 4096 colours.

In the 256 colour modes, the problem is a little more tricky. It is obviously not possible for the dump to create 256 different shades of grey. The way it handles the problem is just to look at the bottom 6 bits of the colour number at any given pixel on the screen, and represent them by the shade held in the shade matrix. This can produce quite useable results, but because the dump detects the colour of a given screen pixel using the POINT function, it is not able to distinguish between the four possible tints of any given colour.

PROGRAM NOTES

The program works in all graphics modes, and the mode selected is automatically set when each screen is loaded. It will not work in modes 3, 6, 7, 11, 14, 16 or 17, which are text only modes. If your printer is not set to produce line feeds, you will need to alter line 350 to read: 350 NEXT:NEXT:VDU1,13,1,10.

All screens are boxed before they are printed out. If you do not require the boxing, remove lines 800 and 810. Because of the way in which dot matrix printers handle graphics printing, the printer may hang if the program stops in the middle of printing a horizontal line. If this happens, press Escape, then turn off the printer at the mains to clear it. Once you have debugged your program of typing errors, this problem should not occur again, because the Escape key is handled by a special routine.

```
   10 REM >TonedumpL
   20 REM Program   Printer Dump
   30 REM Version   A 1.0L
   40 REM Author    Lee Calcraft
   50 REM Risc User December 1987
   60 REM Program   subject to copyright
   70 :
   80 ON ERROR MODE12:REPORT:PRINT" at l
ine ";ERL:OSCLI("FX200"):END
   90 DIM D%(3),shade%(15):*POINTER
  100 REPEAT
  110 PROCdata
  120 MODE12:*CAT
  130 INPUT "FILENAME ";file$
  140 OSCLI("SCREENLOAD "+file$)
  150 PROCpointer
  160 PROCdump
  170 UNTIL FALSE
  180 :
  190 DEFPROCdump
  200 *FX200,1
  210 VDU2,1,27,1,65,1,8
  220 FOR Y%=Y1% TO Y2% +15 STEP -16
  230 VDU1,27,1,76,1,horizdots MOD 256,1
,horizdots DIV 256
  240 FOR X%=X1% TO X2% STEP4
  250 FOR Z%=0 TO 3
  260 Q%=POINT(X%,Y%-Z%*4)AND&F
  270 R%=POINT(X%+2,Y%-Z%*4)AND&F
  280 IFQ%=R% THEN D%(Z%)=shade%(Q%) ELS
E D%(Z%)=(shade%(Q%) AND %111000)+(shade
%(R%) AND %111)
  290 NEXT
  300 FOR R%=2 TO 0 STEP-1
  310 data%=0
  320 FOR Z%=0 TO 3
  330 data%=data%+(((D%(Z%)>>(R%*2))AND%
11)<<(6-2*Z%))
  340 NEXT:VDU1,data%
  350 NEXT:NEXT:VDU1,13
  360 IF INKEY(-113) THEN VDU1,27,1,50,1
,12,3:OSCLI("FX200"):MODE12:PRINT"Escape
":END
```

→

```
370 NEXT:VDU1,27,1,50,1,12,3:*FX200
380 ENDPROC
390 :
400 DEFPROCdata
410 RESTORE
420 FOR A=0 TO 15
430 READ A$:shade%(A)=EVAL("%"+A$)
440 NEXT
450 CASE MODE OF
460 WHEN 0,4,5,18:white=1
470 WHEN 1,8,19:white=3
480 WHEN 2,9,12,20:white=7
490 WHEN 10,13,15:white=15
500 ENDCASE
510 shade%(white)=%111111
520 DATA 000000,100000,000110
530 DATA 100001,100010,011000
540 DATA 100100,100110,010110
550 DATA 101001,111001,110011
560 DATA 111100,011011,011111
570 DATA 111011
580 ENDPROC
590 :
600 DEFPROCpointer
610 MOUSE ON:*FX9
620 REPEAT
  630 IF white=15 THEN GCOL 3,63 ELSE GC
OL 3,7
  640 REPEAT MOUSE P,Q,R:UNTIL R=0
  650 REPEAT MOUSE MX,MY,B:UNTIL B=4
  660 X1%=4*(MX DIV 4)
  670 Y1%=MY-((MY+1) MOD 16)
  680 REPEAT MOUSE P,Q,R:UNTIL R=0
  690 REPEAT
  700 MOUSE MX,MY,B
  710 MX=4*(MX DIV 4)
  720 MY=16*(MY DIV 16)
  730 RECTANGLE X1%,Y1%,MX-X1%,MY-Y1%
  740 RECTANGLE X1%,Y1%,MX-X1%,MY-Y1%
  750 UNTIL B=4
  760 X2%=MX:Y2%=MY:ok=TRUE
  770 IF X1%>=X2% OR Y1%<=Y2% THEN ok=FA
LSE:VDU7
  780 UNTIL ok
  790 IF white=15 THEN GCOL 0,63 ELSE GC
OL 0,7
  800 RECTANGLE X1%,Y1%,MX-X1%,MY-Y1%
  810 RECTANGLE X1%+2,Y1%,MX-X1%,MY-Y1%
  820 horizdots=3+(X2%-X1%)*3/4
  830 ENDPROC
```

SCREEN SAVING

How to save Archimedes graphics screens to disc, and how to do it the quick way.

The Archimedes provides two special commands for saving and loading graphics screens to and from disc. They are:

```
        *SCREENSAVE filename
        *SCREENLOAD filename
```

The save command operates on the currently defined graphics window, or the whole screen if none is defined. The operation of both commands is quite slow for a full screen, because both mode and colour information are saved at the start of the file, preventing a fast block transfer operation from being used by the two commands.

When a screen is loaded back in using *SCREENLOAD, the Archimedes automatically sets both mode and physical colours to match those in use at the time that the screen was saved. It fails on just one point, though: flashing colours. If you get flashing colours after a *SCREENLOAD, that were not on the original screen, use *FX9 to quell them (by removing the second colour).

Of course, you can save and load screens much more quickly using *SAVE and *LOAD; but you will lose the mode and colour information. If this is no problem, use:

```
        *SAVE filename 1FD8000 + length
```

where length=28000 in 160K modes, 14000 in 80K modes, and A000 in 40K modes. Reload with *LOAD filename. This only works when SCREENSIZE is set to 20 - more on this next month. L.G.C.

ACORNSOFT ANSI C

Reviewed by David Pilling

ANSI C is Acornsoft's C compiler for the Archimedes. C is currently the language for software development on micros. The growing libraries of C programs, both in the private and public domain, might mean that C has entrenched itself enough to be the dominant microcomputer language for the foreseeable future. Acorn may well believe this, since ANSI C is one of their first products for the Archimedes.

USING C

Created by Dennis Ritchie, and with its roots in BCPL, C is a relatively primitive high level language, closely related to assembly language. It is easy going, and will allow you to do things without worrying about whether what you are doing makes sense. This should be contrasted with languages like Pascal which have very strict rules about what can and cannot be done. For a long time the only definition of the language was that given in the book on C by Dennis Ritchie and Brian Kernighan (K&R). However, there is now an ANSI standard for the language, and Acorn have adopted this for their new product.

The importance of C to software developers is that an application, once written for one machine, can be easily transferred to another. Certainly, we can expect to see some major programs appearing for the Archimedes which were originally written in C for other machines. ANSI C will be a vital tool for the companies involved.

Acornsoft's ANSI C is supplied on a single 3.5" disc, which also contains all the necessary libraries, including ones for accessing the Archimedes operating system and driving the WIMPs manager. The slim accompanying manual tells the expert C programmer most of what he needs to know. But novices will require a good introductory text on C, as well as the obligatory copy of K&R.

The ANSI C compiler translates a program from one language (C) to another (ARM machine code). Therefore at least two types of files are involved, source code files, which contain C programs, and files containing machine code. ANSI C keeps these files in two separate directories C and P. In the process of creating runnable ARM machine code from C source code, ANSI C generates an intermediate code, somewhat confusingly called ***Object Code***. This code is stored in the O directory of the disc. In addition, the compiler is told through header files (kept in the H directory), the definitions of the functions in the libraries. These libraries are ***linked*** in when the so-called object code is converted to runnable code in the final stage of the compilation process.

COMPILING A PROGRAM

There is an example C program printed below (the PCW Intmath benchmark). Notice the #include statement at the start, which tells the compiler to read the header file for the standard input/output library (stdio.h). To compile this program using ANSI C you would first use a text editor to enter it.

```
/* int math */
#include <stdio.h>
main()
{
int i,x,y;
x=0;y=9;
printf("Start n");
for(i=0;i<1000;i++)
x=x+(y*y-y)/y;
printf("Finish %d n",x);
}
```

C source code for PCW benchtest on Integer Mathematics

This means that you must have a copy of Twin or something similar. Next, the program can be compiled by typing:

```
cc filename
```

During compilation, all referenced header files are read from the H directory; and when compilation is complete, the resultant object code is written to the O directory. Finally, the program must be linked using the standard Acorn linker supplied with ANSI C. This combines the object code file with the object form of the standard libraries to create stand-alone runnable code. This, as we have said, is then stored in the P directory. You can now run your program by typing:

```
*RUN P.filename
```

Acornsoft ANSI C

The ANSI C compiler may also be run using the special form:

```
   cc filename -link
```

This automatically carries out the linking process. Unfortunately, it seems to attempt the link step even if serious errors have occurred in compilation.

ANSI C uses the traditional edit, compile, link and load sequence familiar to generations of programmers. Recently however, languages like Turbo Pascal have made popular the idea of having all these stages resident in the machine at once. In effect the compiler has a built-in editor from which you can give an instruction to run your program. If any errors are detected, the editor is automatically re-entered at the offending line. The complete cycle from edit to run can thus take place without disc access. It is a pity that these features are not available on ANSI C.

BENCHMARKS

The accompanying table shows the PCW benchmarks for ANSI C compared to those for Archimedes Basic V, Model B Basic II and Turbo C on a PC. The results are quite impressive, the overall PCW rating for the Archimedes being pushed up to 20; towards the minicomputer class. The worst benchmarks are those involving floating point and trig functions. This may be a result of ANSI C using the floating point emulator, a copy of which is supplied on the C disc.

ANSI C produces fully stand-alone object code. This is however, somewhat voluminous, including as it does a standard run time library of some 30K. Without the library, each PCW benchmark would compile to about 200 bytes of ARM machine code; though of course, each piece of object code will need certain parts of the library in order to run.

In fact the ANSI C compiler is itself written in C, and is around 250K in length. Owners of A305 machines should note that ANSI C needs about 1M byte of memory to run. Disc space is also a bit tight, unless you have dual drives or a Winchester. Stripping all the unnecessary files off the C disc and adding a copy of Twin, produces a disc with 270K of free space. This is probably enough for a small to medium sized project. You must remember however, that this space has to hold source, object and machine code versions of your program. Serious development work will either need a second floppy or a hard disc drive. A440 owners will have more room in which to manoeuvre. I found that I could only just get all the PCW benchmarks on a single disc along with the compiler.

	Int	Real	Trig	Text	Graf	Store	Index
ANSI C	0.024	0.278	3.79	4.50	5.55	3.79	20.84
Basic V	0.370	0.410	2.31	6.67	6.78	6.71	11.03
Basic II	2.600	5.800	76.50	15.20	21.10	23.50	1.67
Turbo C	0.045	3.700	26.10	72.30		12.80	4.67

All timings in this table are in seconds except for Index. This is PCW's new speed index, giving an average rating for a given system, in which a high index means a fast performance.

Compiling some C files which I had previously used with no problem on a PC, I was horrified to see dozens of error messages appear. ANSI C is very particular about what is legal and what is not. Things which other compilers turn a blind eye to, ANSI C points out. At first this appears to be a disadvantage. However, after a while you get used to it, and no doubt it improves the code you write. Errors come in three categories: warnings, errors and serious errors. Some of the non-serious ones can be turned off, making it easier to spot the important ones. Indeed, the number and quality of the error messages in this package is a credit to the authors A.C.Norman and A.Mycroft.

To summarise, ANSI C is a useful and well written product. For the serious Archimedes programmer it is, with Twin and the Acorn assembler, a necessity. But no doubt, it will not be the last word in C compilers for the Archimedes.

ANSI C, £113.85 inc VAT
Supplied by:
Acorn Computers Ltd.,
Cambridge Technopark,
645 Newmarket Rd,
Cambridge CB5 8PD.

RU

CONFIGURING ARCHIMEDES

Lee Calcraft takes a look at the *CONFIGURE command, and its implications, and presents a program to display and save to disc the precious contents of your CMOS RAM.

As you may know, the Archimedes has 240 bytes of CMOS non-volatile RAM. This means that when you switch your machine off, although the contents of the normal RAM are lost, CMOS RAM retains its data. The computer uses this CMOS RAM for a number of different purposes, including the `*CONFIGURE` options. We will first of all take a look at the way in which these work, and then present a program for displaying and saving away the entire contents of CMOS RAM so that when your machine gets wrongly configured, you can just load the values back in from disc.

CONFIGURING YOUR SYSTEM

If you type `*STATUS` you will see a display of the current configuration settings of your machine. For example if you look at the number against the word *Mode* this tells you which screen mode is selected at power up. By default it is mode 0: to change it to mode 4 say, just type:

```
    *CONFIGURE MODE 4
```

This information will be stored in CMOS RAM, and when the machine is next powered up, or Ctrl-Break is pressed (hold down the Ctrl key, and tap Break), the desired mode will be selected.

RAM ALLOCATION

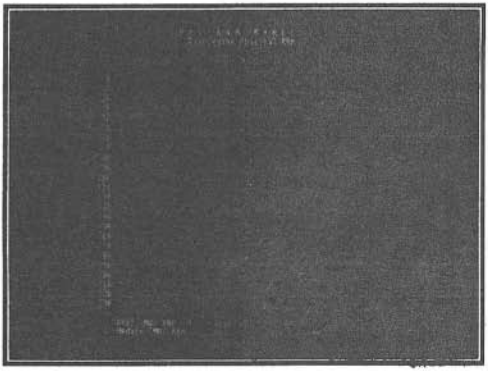
A number of these configure options are used to determine the way in which the computer's main memory is allocated. They allocate how much RAM is set aside for the screen, for relocatable modules, and so on. The fifth item allocates RAM to a RAM filing system but this is not yet available.

ScreenSize	**Screen RAM**
RMASize	**Relocatable Module RAM**
SpriteSize	**Sprite RAM**
FontSize	**Font cache allocation**
RAMFsSize	**RAM Filing System**
SystemSize	**System RAM**

When each machine leaves the factory, its CMOS RAM is initialised with a set of default values. The default state of the 305 and 310 machines differ in that while the 305 has 80K allocated for screen memory, the 310 has double this figure. This means that in its default state, the 305 cannot use modes 15, 16, 17, 19 or 20. But it is a simple matter to reconfigure it. Just execute:

```
    *CONFIGURE SCREENSIZE 20 or *CON.SC.20
```

then perform a Ctrl-Break to bring the option into effect. This configures a block of 20 pages of 8K (making a total of 160K) for screen use. To change back to the default at any time, just repeat the sequence with *CON.SC.0



The RMA size is also allocated in 8K pages. But Acorn advise that it is best to keep the RMA allocation as low as possible. Even if you set it to zero with:

```
    *CON.RMAS.0     (then press Ctrl-Break)
```

you can still load as many relocatable modules as you wish, providing that you are not in an application such as Basic. So to load in a module, just type:

```
    QUIT                   (to leave Basic)
    *RMLOAD modulename     (load module)
    BASIC                  (re-enter Basic)
```

The quantity of RAM allocated to sprites is controlled in a similar way. Thus to reserve 48K for sprite space, use:

```
    *CON.SPR.6     (then press Ctrl-Break)
```

The reserving of RAM for so-called font caches follows the same pattern, except that the page size is 4K rather than 8. To allocate 16K for font use, type:

```
    *CON.FONT.4    (then press Ctrl-Break)
```

THIRD PARTY CONFIGURING

In order to make their software work correctly on all Archimedes systems, however configured, some pieces of software currently on the market reconfigure the machine they are running on. In most cases the software also attempts to reset the machine to its original state after use, but this does not always work, as for example when the user turns off his machine without going through the expected exit protocol of the software concerned. To take a practical example, depending how you quit Zarch, your machine can be left configured with no window manager (and therefore no way of turning on the mouse pointer), and with the machine no longer starting up in Basic.

In a recent directive, Acorn has urged all third party suppliers of software not to automatically reconfigure host computers, but to explain to the user the need for reconfiguring, and how to perform it. In the mean time, for those who have lost their pointer, execute:

```
    *RMREINIT WindowManager
```

then press Ctrl-Break.

INSTANT DEFAULT

If your machine does get deconfigured at any time, you can reconfigure it to its factory default settings as follows. Turn the power on while pressing the "R" key, then turn off again, and repeat the process. You need to perform the power-up twice, because with each power-up, the monitor type is toggled between normal and multi-sync. One snag with this trick is that it resets the system clock to January 1900. Moreover, it resets *all* options to their default, and this may prove extremely inconvenient.

RISC USER CMOS RAM MANAGER

Because the CMOS RAM settings, including those serviced by `*CONFIGURE` and `*SHOW`, are so easily lost or interfered with, we present here a short utility to help you to take control over the situation more effectively. When the accompanying program is run, it will display the values held in the Archimedes' 240 bytes of CMOS RAM. The data is colour-coded to indicate Acorn's latest allocation of this RAM.

COLOUR	BYTES	USE
Cyan	0-16	MachineConfiguration
Green	17-29	Reserved for Acorn
Yellow	30-45	For the User
Magenta	46-111	Applications Software
Grey	112-239	Operating System Use

A menu at the bottom of the display offers six options. The first reads and displays the contents of CMOS RAM. The third allows the displayed data (whatever its source) to be saved to disc. Any number of data files may be created to reflect different machine configurations set up for different purposes. The second option permits files to be reloaded and displayed; though without affecting the current contents of CMOS RAM. A fourth option allows the machine's CMOS RAM to be updated with the currently displayed settings.

Finally, a feature is incorporated to assist in comparing various settings. Each time that data is displayed, comparisons are made with the set of data previously displayed, and any item which has altered is flagged with an asterisk. This allows you to compare the current CMOS RAM setting of your machine with another stored on disc, or to compare the values held in two disc files. Lastly, in order to help you to keep track of the data currently displayed, its source (whether physical CMOS RAM, or a disc file) is displayed at the top of the screen.

PROGRAM NOTES

CMOS RAM is written to using *FX162:

```
    *FX162,RAMno,data
```

This puts the value *data* into RAM number *RAMno*. Note that only bytes 1 to 239 may be written to: byte zero, the Econet station number, is write-protected.

CMOS RAM is read using OSBYTE 161:

```
    SYS 6,161,RAMno TO X, Y, Z
```

Where the variable Z returns the value held in RAM number *RAMno*.

```
   10 REM           >CmosC
   20 REM Program   CMOS RAM Manager
   30 REM Version   A 1.0C
   40 REM Author    Lee Calcraft
   50 REM Risc User December 1987
   60 REM Program   Subject to copyright
   70 :
   80 ON ERROR MODE 12:@%=&90A:REPORT:PR
INT" at line ";ERL:END
   90 DIM cmos(240),oldmos(240)
  100 MODE12:COLOUR128,0,0,200:@%=&304
  110 COLOUR 8,170,170,170:Q$=" "
  120 PROCreadram
  130 REPEAT
  140 PROCdisplay
  150 REPEAT:M=FNmenu
  160 UNTIL (M>48 AND M<54) OR M=42
  170 CASE M OF
  180 WHEN 42:PROCos
  190 WHEN 49:PROCreadram
  200 WHEN 50:PROCreaddisc
  210 WHEN 51:PROCwritedisc
  220 WHEN 52:PROCupdate:PROCreadram
  230 ENDCASE
  240 UNTIL M=53:@%=&90A:END
  250 :
  260 DEFPROCreadram
  270 display$="Physical RAM"
  280 REM oldmos()=cmos()
  290 FOR A=0 TO 239
  300 SYS 6,161,A TO X,Y,cmos(A)
  310 NEXT:ENDPROC
  320 :
  330 DEFPROCdisplay
  340 CLS:PRINTTAB(25,0)"C M O S   R A M
   M A N A G E R"
  350 PRINTTAB(29)"Displaying ";display$
  360 PRINTTAB(15,3);
  370 FOR B=0 TO 9:PRINTB;" ";:NEXT:PRIN
T'
  380 FOR A=0 TO 23:PRINTSPC5,A*10:NEXT
  390 COLOUR6:PRINTTAB(0,8);
  400 FOR A=0 TO 23
  410 PRINTTAB(15,5+A);
  420 FOR B=0 TO 9
  430 N=B+10*A
  440 IF N=17 COLOUR2
  450 IF N=30 COLOUR3
  460 IF N=46 COLOUR5
  470 IF N=112 OR N=135 OR N=148 COLOUR8
  480 IF N=134 OR N=143 COLOUR7
  490 P$=" ":IF cmos(N)<>oldmos(N) P$=Q$
  500 PRINTcmos(N);P$;
  510 NEXT:NEXT
  520 LINE 100,880,1100,880
  530 LINE 220,920,220,100
  540 oldmos()=cmos()
  550 Q$="*":COLOUR7:PROCkey:ENDPROC
  560 :
  570 DEFFNmenu
  580 PRINTTAB(10,30)"1. Read CMOS RAM
   2. Read Disc     3. Write Disc"
  590 PRINTTAB(10,31)"4. Update CMOS RAM
   5. Quit          * O.S. Command ";
  600 OSCLI("FX15"):=GET
  610 :
  620 DEFPROCreaddisc
  630 VDU28,7,31,65,30:COLOUR136:CLS
  640 INPUT "Filename for Read ",file$
  650 CLOSE#0:D%=OPENIN(file$)
  660 oldmos()=cmos()
  670 FOR A=0 TO 239
  680 cmos(A)=BGET# D%
  690 NEXT
  700 CLOSE#D%:VDU26:COLOUR128
  710 display$="Disc File "+file$
  720 ENDPROC
  730 :
  740 DEFPROCwritedisc
  750 VDU28,7,31,65,30:COLOUR136:CLS
  760 INPUT"Filename for Save ",file$
  770 CLOSE#0:D%=OPENOUT(file$)
  780 FOR A=0 TO 239
  790 BPUT#D%,cmos(A)
  800 NEXT
  810 CLOSE#D%:VDU26:COLOUR128
  820 ENDPROC
  830 :
  840 DEFPROCupdate
  850 FOR A=1 TO 239
  860 OSCLI("FX162,"+STR$(A)+","+STR$(cm
os(A)))
  870 NEXT:ENDPROC
  880 :
  890 DEFPROCos
  900 VDU28,7,31,65,30:COLOUR136:CLS
  910 INPUTLINE"*"command$
  920 VDU26:CLS:OSCLI(command$)
  930 PRINT'"Press any key ";:A=GET
  940 COLOUR128:ENDPROC
  950 :
  960 DEFPROCkey
  970 PRINTTAB(66,17)"KEY:"
  980 PRINTTAB(66,18)"134 Fonts"
  990 PRINTTAB(66,19)"143 Screen"
 1000 PRINTTAB(66,20)"144 RAM FS"
 1010 PRINTTAB(66,21)"145 System"
 1020 PRINTTAB(66,22)"146 RMA"
 1030 PRINTTAB(66,23)"147 Sprites"
 1040 ENDPROC
```

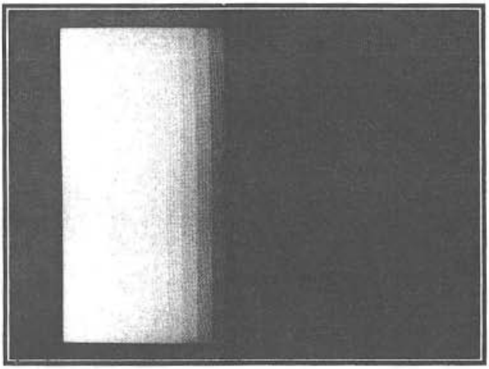
RU

ARCHIMEDES VISUALS

This month's visual effects include a cylinder drawing routine using a pixel-based shading technique, a textured background and a multi-coloured sketch routine - all using a minimum of code.

SUBTLY SHADED CYLINDERS

Last month's "Visuals" was built around a sphere drawing routine written by Paul Fellows at Acorn. It used filled circles drawn in 8 different shades to produce quite an effective three dimensional image. This month we are featuring a program from Tim Dobson at Acorn which approaches the problem of three dimensional shading in a different way.



Program Cyanafade

The chief drawback with the method used last month is that the shading does not appear to be continuous. There is a very noticeable step as the shades change. Listing 1 demonstrates this effect. It generates a series of cyan bands of progressively darker shade; and as you will see when you run the program, there is no sense of continuity.

Listing 1

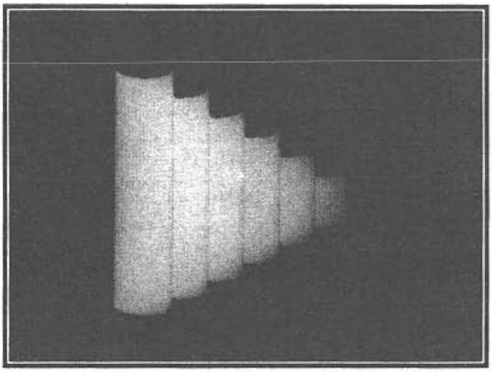
```
   10 REM >Cyanfade2
   20 REM Grey Scale in Cyan
   30 MODE 12
   40 FOR Z=1 TO 15
   50 COLOUR Z,0,Z*16,Z*16
   60 GCOL Z
   70 RECTANGLEFILL 200,0,50*(16-Z),1023
   80 NEXT:COLOUR15
```

Tim Dobson's program attempts to get around this by creating shaded surfaces one pixel at a time. This allows him to incorporate a random component in the pixel shade which softens the effect. You will get the idea if you run listing 2. The first few lines set up 8 shades of yellow in colours 8 to 15. PROCcylinder then uses the POINT function to draw a small cylinder one pixel at a time. Considering the work involved, this is accomplished relatively quickly. PROCcopy is then called, which uses the Archimedes' excellent rectangle copy routine to finish off the picture at great speed.

Listing 2

```
   10 REM >Cylinder4
   20 REM Program   Set of Cylinders
   30 REM Version   A 0.4
   40 REM Author    Tim Dobson
   50 REM Risc User December 1987
   60 REM Copyright Acorn Computers
   70 :
   80 MODE12:OFF
   90 DIM COL%(7):A=RND(-1000)
  100 FORI%=0TO7:COL%(I%)=8+I%:NEXT
  110 FORI%=8TO15:VDU19,I%,16,I%<<4,I%<<
4,NOT(I%<<4):NEXT
  120 VDU19,8,16,&60,&60,0
  130 VDU19,9,16,&80,&80,0
  140 PROCcylinder
  150 PROCcopy
  160 ON:END
  170 :
  180 DEFPROCcylinder
  190 R%=100:R2%=R%*R%
  200 K1=4.2:K2=2.3:XC%=800
  210 VDU29,XC%+200;512;:YS%=80
  220 FORX%=R%TO0STEP-2
  230 X2%=X%*X%
  240 COL=K1*SQR(1-X2%/R2%)
  250 COL2=K2-COL/2
  260 int%=COL2:fra=COL2-int%
  270 Y%=SQR(R2%-X2%)/3
  280 FORY1%=YS%-Y%TOY%+YS%STEP4
  290 GCOLCOL%(int%-(RND(1)<=fra))
```

```
300 POINTX%,Y1%:POINT-X%,Y1%
310 NEXT
320 COL2=K2+COL
330 int%=COL2:fra=COL2-int%
340 FORY1%=-YS%-Y%TO YS%-Y%STEP4
350 GCOLCOL%(int%-(RND(1)<=fra))
360 POINTX%,Y1%:POINT-X%,Y1%
370 NEXT:NEXT:ENDPROC
380 :
390 DEFPROCcopy
400 FORJ%=1TO5
410 MOVE -R%,-YS%-Y%:MOVE R%,YS%+Y%
420 PLOT 190,-R%-120,-YS%-Y%-60
430 XC%=XC%-120:VDU29,XC%+200;512;
440 MOVE -R%,YS%-Y%-180
450 MOVE R%,YS%+Y%-60
460 PLOT 190,-R%,YS%-Y%-60
470 YS%=YS%+60
480 NEXT:ENDPROC
```



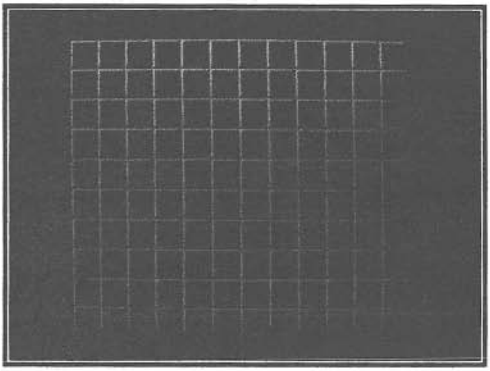
Program Cylinder

If you want to see how the copy procedure creates the larger cylinders from the small one, add the following lines, which will halt the program after each replication, until a key is pressed:

```
405 A=GET
435 A=GET
```

TEXTURES FROM PAINTBOX

You will probably be familiar with some of the impeccable visual effects that the television companies have created using Quantel's Paintbox. Listing 3 attempts to simulate one very simple Quantel-generated effect. It is a textured background which was used in BBC TV's coverage of Wimbledon '87. The program works by creating a single 3D object in the bottom left hand corner of the screen (PROCbox). Then PROCcopy replicates this 119 times.



Program Texture

Listing 3

```
 10 REM >Texture4
 20 REM Textured Surface
 30 REM by Lee Calcraft
 40 MODE 12:COLOUR 8,0,128,48
 50 COLOUR 9,0,80,8
 60 COLOUR 10,32,160,80
 70 PROCbox
 80 PROCcopy
 90 END
100 :
110 DEFPROCbox
120 GCOL 9:RECTANGLE FILL 0,0,100,100
130 GCOL 10:MOVE 0,0:MOVE 0,100
140 PLOT 85,100,100
150 GCOL 8:RECTANGLE FILL 10,10,80,80
160 ENDPROC
170 :
180 DEFPROCcopy
190 MOVE 0,0:MOVE 100,100
200 FOR X%=0 TO 1100 STEP 100
210 FOR Y%=0 TO 900 STEP 100
220 MOVE 0,0:MOVE 100,100
230 PLOT 191,X%,Y%
240 NEXT:NEXT:ENDPROC
```

Continued on page 26

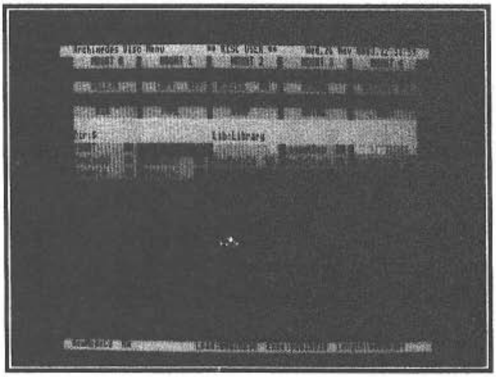
MENU

ARCHIMEDES DISC MENU (Part 2)

By David Pilling

Append this code to last month's disc menu program to give a whole host of extra features. These include the operations of Copy, Delete, Rename and Settype on marked files, improved error handling and fuller file information.

The response to last month's menu has been encouraging, with just one small niggle. If you have operating system 0.3 then you need to preface the "FFA" in line 2910 with an ampersand (&). The published version works without alteration on both the 0.2 and the new series one operating systems.



The program listing accompanying this article, as promised, adds a number of useful features to the menu. To get it working, you can either type it in with last month's program already in the machine, or you can type the listing into a cleared machine, and proceed as follows. First of all, save this month's listing to disc for safety. Then type the following from Basic:

```
    *SPOOL upgrade
    LIST
    *SPOOL
```

Then load in last month's program, and type:

```
    *EXEC upgrade
```

Ignore the reported syntax errors, and you will now have in your machine a merged version of the two programs. You cannot use the Archimedes' APPEND command, because the two listings have overlapping line numbers.

When you run the new program, it will create a new relocatable module called RMENU, and this can be installed as described last month. Then use *MENU, or the new command *M to call it. As you will now see, the menu contains eight extra choices in the upper area. These new options are all carried out on marked files. And you will see that if you perform a single click on any of the file or directory names, it will be highlighted. A second click will unmark the item, or alternatively, you may use the **UNTAG** box to unmark all files. Untagging and all other actions are initiated by double clicking. The remaining seven options operate as follows:

DELETE - This will delete all marked files.

COPY - This allows the copying of all marked files to a destination directory supplied by the user. Note that no destination filename is required. As currently set, the copy option works in **QPF** mode. That is to say, Quick (i.e. it uses the whole of RAM), Prompting (i.e. it assumes that you have only one disc drive, and prompts for disc changes) and Forced (i.e. it will force the overwriting of locked files). But these options can be toggled on or off by single clicking the appropriate mouse button, when the pointer is over the COPY box. Note that because of the way in which the Archimedes *COPY command has been rewritten on operating system series one, the Quick option leaves you in Arthur, and prevents multiple file copying.

ACCESS - This permits the changing of access status to that in the ACCESS box. Again you can toggle the setting by single clicking the mouse buttons when the pointer is on this box.

SETTYPE - With this option you can change the file type of all marked files to the type appearing in the SETTYPE box. Its default setting is FFE, but may be altered by single-clicking with the two outer mouse buttons.

RENAME, TYPE and DUMP - These final options perform a *RENAME, *TYPE or *DUMP on all marked files.

NOTE: Operating system series one users should add the following line to last month's program to compensate for Y coordinate changes in the new operating system:

```
    1865 ADD R1,R1,#16
```

This month's magazine disc contains a complete version of the full feature menu. More on using the menu next month.

```
    5 REM >MenuDiff3
   80 DIM code &3000:PROCsetup
  171 EQUS"M":EQUB0:ALIGN
  172 EQUD go-code:EQUD0:EQUD0
  173 EQUD helpt-code
  315 BL eset:BL tagcl
  352 BL copp
  356 BL accp:BL stpp
  500 MOV R12,#0
  510 B loopx
  520.loop
  530 BL prifx
  685 B knam
  686.knam2
  725.knam3
  840 ADR R0,blok:SWI cli
  842.rsx:SWI ws
  895 MOV R5,R2:MOV R6,R0:MOV R7,R1
 1330 ADR R0,dir:SWI cli:B untag
 1450 B comz
 1610 ADR R0,mout:SWI cli:B untag
 1720 SWI cli:B untag
 1730.msing:MOV R0,R6:MOV R1,R7:BL maps
 1731 BEQ mloop:CMP R0,#&10:BEQ cops
 1732 CMP R0,#&11:BEQ accs:CMP R0,#&13
 1733 BEQ sxx:MOV R0,R6:MOV R1,R7
 1734 BL maps:SUBS R0,R0,#&28:BMI mloop
 1735 CMP R0,R11:BGE mloop:MOV R12,R0
 1736 BL tass:EOR R4,R4,#1
 1737 STRB R4,[R5,R12]
 1740 BL prifx:B mloop
 1780.quit2:BL erset:SWI os
 1781 ADR R0,escp
 1790 SWI gen
 1795.escp
 1796 EQUD17:EQUS"Escape":EQUB0
 2831 EQUB31:EQUB3:EQUB3
 2832 EQUS"    DELETE    "
 2837 EQUB31:EQUB48:EQUB3
 2840 EQUS"    RENAME    "
 2841 EQUB31:EQUB18:EQUB5
 2842 EQUS"    TYPE      "
 2843 EQUB31:EQUB33:EQUB5
 2844 EQUS"    DUMP      "
 2845 EQUB31:EQUB48:EQUB5
 2846 EQUS"    UNTAG     "
 2849 EQUB31:EQUB63:EQUB5
 2890 ]:P%=P%+&1000:PROCADD:NEXT
 2900 OSCLI("SAVE RMENU "+STR$~code+" "+
STR$~(end))
 2985 con=&F:cnv=&C1:gen=&2B
 3000 DEFPROCADD
 3010 [ OPT pass
 3020.tags:EQUS CHR$1+STRING$(80,CHR$0)
 3030.tass:ADR R5,tags:LDRB R4,[R5,R12]
 3040 CMP R4,#1:MOV R15,R14
 3050.acc:EQUB 6
 3060.cob:EQUB 7:ALIGN
 3070.styp:EQUD &FFE
 3080.tasq:MOV R6,R14
 3090.tasq1:BL tass:BEQ tasq2
 3100 CMP R12,#78:MOVEQ R15,R6
 3110 ADD R12,R12,#1:B tasq1
 3120.tasq2:BL poin:ADD R12,R12,#1
 3130 CMP R0,#0:MOV R15,R6
 3140.tagcl
 3150 MOV R0,#0:MOV R1,#0:ADR R2,tags
 3160.tagl
 3170 STRB R0,[R2],#1:ADD R1,R1,#1
 3180 CMP R1,#78:BNE tagl:MOV R15,R14
 3190.accs:LDRB R0,acc
 3200 EOR R0,R0,R5:STRB R0,acc
 3210 BL blog:BL accp:B mloop
 3220.sxx:LDR R0,styp:CMP R5,#1
 3230 ADDEQ R0,R0,#1:CMP R5,#4
 3240 SUBEQ R0,R0,#1:STR R0,styp
 3250 BL blog:BL stpp:B mloop
 3260.accq:EQUS"ACCESS "
 3270.accn:EQUS STRING$(16,CHR$0)
 3280.acp:MOV R12,#0
 3290.acl1:BL tasq:BEQ newd
 3300 ADR R0,accn:BL mbsb:SUB R0,R0,#1
 3310 LDRB R1,acc:MOV R2,#32
 3320 STRB R2,[R0],#1:TST R1,#1
 3330 MOV R2,#ASC"L":STRNEB R2,[R0],#1
 3340 TST R1,#2:MOV R2,#ASC"W"
 3350 STRNEB R2,[R0],#1:TST R1,#4
 3360 MOV R2,#ASC"R":STRNEB R2,[R0],#1
 3370 MOV R2,#13:STRB R2,[R0]
 3380 ADR R0,accq:SWI cli:B acl1
 3390.prifx
 3400 MOV R6,R14:BL tabn:BL poin
 3410 LDRB R3,[R1,#&10]:CMP R3,#2
 3420 BEQ lod1
 3430 BL tass:BLNE blog:BLEQ gobl:B lod2
 3440.lod1
 3450 BL tass:BLNE grog:BLEQ gorg
 3460.lod2
 3470 MOV R14,R6:B prif
 3480.typ:MOV R12,#0:SWI wr+22:SWI wr+14
 3490.typl:BL tasq:BEQ rsx
 3500 LDRB R2,[R1,#&10]:CMP R2,#2
 3510 BEQ typl:ADR R0,typn:BL mbsb
 3520 ADR R0,typq:SWI cli:B typl
 3530.typq:EQUS"TYPE "
 3540.typn:EQUS STRING$(12,CHR$0)
 3550.dmp:MOV R12,#0:SWI wr+22:SWI wr+14
 3560.dmpl:BL tasq:BEQ rsx
 3570 LDRB R2,[R1,#&10]:CMP R2,#2
 3580 BEQ dmpl:ADR R0,dmpn:BL mbsb
 3590 ADR R0,dmpq:SWI cli:B dmpl
 3600.dmpq:EQUS"DUMP "
 3610.dmpn:EQUS STRING$(12,CHR$0)
 3620.stpp:SWI ws:EQUB31:EQUB63:EQUB3
 3630 EQUS"  SETTYPE ":EQUB0:ADR R2,sssc
 3640 MOV R3,#3:LDR R1,styp
```

```
3650 MOV R1,R1,ROR#12
3660.styl:MOV R1,R1,ROR#28
3670 AND R0,R1,#15:CMP R0,#10
3680 ADDGE R0,R0,#ASC"A"-10
3690 ADDLT R0,R0,#ASC"0":SWI wc
3700 STRB R0,[R2],#1:SUBS R3,R3,#1
3710 BNE styl:SWI wr+32:MOV R15,R14
3720.sssc:EQUD 0
3730.sccq:EQUS"SETTYPE "
3740.sccn:EQUS STRING$(16,CHR$0)
3750.gorg
3760 SWI wr+17:SWI wr+8:SWI wr+17
3770 SWI wr+128+11:MOV R15,R14
3780.ssxg:MOV R12,#0
3790.sxgl:BL tasq:BEQ newd:ADR R0,sccn
3800 BL mbsb:SUB R0,R0,#1:ADR R2,sssc
3810 MOV R3,#3:MOV R4,#32
3820 STRB R4,[R0],#1
3830.sxgx
3840 LDRB R4,[R2],#1:STRB R4,[R0],#1
3850 SUBS R3,R3,#1:BNE sxgx
3860 MOV R2,#13:STRB R2,[R0]
3870 ADR R0,sccq:SWI cli:B sxgl
3880.gobl
3890 SWI wr+17:SWI wr+8:SWI wr+17
3900 SWI wr+128:MOV R15,R14
3910.accp
3920 SWI ws:EQUB31:EQUB33:EQUB3
3930 EQUS" ACCESS   ":EQUB0:LDRB R0,acc
3940 TST R0,#4:SWIEQ wr+32
3950 SWINE wr+ASC"R":TST R0,#2
3960 SWIEQ wr+32:SWINE wr+ASC"W"
3970 TST R0,#1:SWIEQ wr+32
3980 SWINE wr+ASC"L":SWI wr+32
3990 MOV R15,R14
4000.delq:EQUS"DEL."
4010.deln:EQUS STRING$(12,CHR$0)
4020.del:MOV R12,#0
4030.dell:BL tasq:BEQ untag
4040 LDRB R2,[R1,#&C]:TST R2,#8
4050 BNE dell:ADR R0,deln:BL mbsb
4060 ADR R0,delq:SWI cli:B dell
4070.copp
4080 SWI ws:EQUB31:EQUB18:EQUB3
4090 EQUS"   COPY   ":EQUB0:LDRB R0,cob
4100 TST R0,#4:SWIEQ wr+32
4110 SWINE wr+ASC"Q":TST R0,#2
4120 SWIEQ wr+32:SWINE wr+ASC"P"
4130 TST R0,#1:SWIEQ wr+32
4140 SWINE wr+ASC"F"
4150 SWI wr+32:MOV R15,R14
4160.cops:LDRB R0,cob
4170 EOR R0,R0,R5:STRB R0,cob
4180 BL blog:BL copp:B mloop
4190.copy
4200 SWI ws:EQUB31:EQUB3:EQUB6
4210 EQUS"Enter dest path >"
4220 EQUB23:EQUB1:EQUB1:EQUB0
4230 SWI wr:SWI wr:SWI wr:SWI wr:SWI wr
4240 SWI wr:SWI wr:ADR R0,string
4250 MOV R1,#77:MOV R2,#32:MOV R3,#128
4260 SWI rs:BCS ex
4270 MOV R12,#0
4280 LDRB R0,cob:TST R0,#2:BEQ copl
4290 SWI wr+22:SWI wr+14
4300.copl:BL tasq:BEQ coox
4310 ADR R0,copn:BL mbsb:SUB R0,R0,#1
4320 MOV R2,#32:STRB R2,[R0],#1
4330 ADR R1,string
4340.col3:LDRB R2,[R1],#1
4350 CMP R2,#13:BEQ col4
4360 STRB R2,[R0],#1:B col3
4370.col4
4380 MOV R2,#ASC".":STRB R2,[R0],#1
4390 SUB R12,R12,#1:MOV R5,R0
4400 BL poin:MOV R0,R5:ADD R12,R12,#1
4410 BL mbsb
4420 SUB R0,R0,#1:MOV R2,#32
4430 STRB R2,[R0],#1:LDRB R1,cob
4440 TST R1,#1:MOV R2,#ASC"F"
4450 STRNEB R2,[R0],#1:TST R1,#2
4460 MOV R2,#ASC"P":STRNEB R2,[R0],#1
4470 TST R1,#4:MOV R2,#ASC"Q"
4480 STRNEB R2,[R0],#1
4490 MOV R2,#13:STRB R2,[R0]
4500 ADR R0,copq:SWI cli:B copl
4510.coox:LDRB R0,cob
4520 TST R0,#2:BEQ newd:BNE go2
4530.string:EQUS STRING$(80,CHR$0)
4540.copq:EQUS"COPY "
4550.copn:EQUS STRING$(128,CHR$32)
4560.comz
4570 CMP R0,#&17:BEQ untag
4580 CMP R0,#&A:BEQ del
4590 CMP R0,#&15:BEQ typ
4600 CMP R0,#&16:BEQ dmp
4610 CMP R0,#&C:BEQ acp
4620 CMP R0,#&E:BEQ ssxg
4630 CMP R0,#&B:BEQ copy
4640 CMP R0,#&D:BEQ rename:BNE mloop
4650.renn
4660 EQUS "RENAME "
4670.renq
4680 EQUS STRING$(32,CHR$0)
4690.rename
4700 MOV R12,#0
4710.renl
4720 BL tasq:BEQ untag
4730 LDRB R2,[R1,#&C]:TST R2,#8
4740 BNE renl
4750 ADR R0,renq:BL mbsb:SUB R0,R0,#1
4760 MOV R2,#32:STRB R2,[R0],#1
```

```
4770 MOV R6,R0:SUB R12,R12,#1
4780 BL poin:ADD R12,R12,#1
4790 SWI ws:EQUB31:EQUB3:EQUB6
4800 EQUS STRING$(74,CHR$32)
4810 EQUB31:EQUB3:EQUB6
4820 EQUS"Enter new name for "
4830 EQUB23:EQUB1:EQUB1:EQUB0
4840 SWI wr:SWI wr:SWI wr:SWI wr:SWI wr
4850 SWI wr:SWI wr:BL prif
4860 SWI wr+ASC":":ADR R0,string
4870 MOV R1,#77:MOV R2,#32:MOV R3,#128
4880 SWI rs:BCS ex
4890 ADR R1,string:MOV R0,R6
4900 LDRB R2,[R1]:CMP R2,#13:BEQ renl
4910.ren2
4920 LDRB R2,[R1],#1:STRB R2,[R0],#1
4930 CMP R2,#13:BNE ren2
4940 ADR R0,renn:SWI cli:B renl
4950.oerrorh:EQUD 0
4960.oerbuff:EQUD 0
4970.eset
4980 ADR R0,errorh:ADR R1,erbuff
4990 MOV R2,#0:MOV R3,#0
5000 SWI con
5010 STR R0,oerrorh:STR R1,oerbuff
5020 MOV R15,R14
5030.erbuff
5040 EQUS STRING$(128,CHR$0)
5050.errorp
5060 ADR R1,erbuff+8:MOV R2,#0
5070 LDRB R0,[R1]:CMP R0,#0:BEQ errpx
5080 ADR R1,erbuff+8:MOV R2,#0
5090.errorl
5100 LDRB R0,[R1]:CMP R0,#0
5110 STRB R2,[R1],#1
5120 SWINE wc:BNE errorl
5130.errpx
5140 MOV R15,R14
5150.erset
5160 LDR R0,oerrorh:LDR R1,oerbuff
5170 MOV R2,#0:MOV R3,#0
5180 SWI con
5190 MOV R15,R14
5200.errorh
5210 SWI ws:EQUB28:EQUB10:EQUB12
5220 EQUB70:EQUB6:EQUB17:EQUB129
5230 EQUB17:EQUB7:EQUB12:EQUB10
5240 EQUS" Error !":EQUB31:EQUB1
5250 EQUB3:EQUB7:EQUB0
5260 BL errorp
5270 SWI ws:EQUB31:EQUB1:EQUB5
5280 EQUS"Click to continue":EQUB0
5290.errcl:SWI es:BCS ex
5300 SWI ms:ANDS R2,R2,#7:BEQ errcl
5310.errc2
5320 SWI ms:ANDS R2,R2,#7:BNE errc2
5330 B ex
5340.knam:LDR R0,[R1]
5350 ADDS R0,R0,#&100000:BCS knam0
5360 SWI ws:EQUS"          "
5370 EQUB 0:B knam2
5380.knam0:LDR R0,[R1]
5390 MOV R0,R0,LSR#8:BIC R0,R0,#&FF0000
5400 BIC R0,R0,#&F000:MOV R4,R0
5410 MOV R3,#3:MOV R2,R0,ROR#12
5420 SWI wr+32
5430.knal
5440 MOV R2,R2,ROR#28
5450 AND R0,R2,#15:CMP R0,#10
5460 ADDGE R0,R0,#ASC"A"-10
5470 ADDLT R0,R0,#ASC"0":SWI wc
5480 SUBS R3,R3,#1:BNE knal
5490 SWI wr+32
5500 AND R0,R4,#&F00:CMP R0,#&F00
5510 MOVNE R4,#0:AND R0,R4,#&FF
5520 ADR R5,ttab:MOV R3,#10
5530.ktyl:LDRB R2,[R5],#1
5540 CMP R2,#0:BEQ ktyll
5550 CMP R2,R0:BEQ ktyll
5560 ADD R5,R5,#10:B ktyl
5570.ktyll:LDRB R0,[R5],#1:SWI wc
5580 SUBS R3,R3,#1:BNE ktyll:SWI wr+32
5590 LDR R0,[R1],#4:AND R0,R0,#&FF
5600 MOV R6,R1:STRB R0,kblk1
5610 LDR R0,[R1]:STR R0,kblk2
5620 ADR R0,kblk2:ADR R1,kstr
5630 MOV R2,#30:ADR R3,kstr2
5640 SWI cnv:ADR R1,kstr
5650.kmlp
5660 LDRB R0,[R1],#1:CMP R0,#0
5670 SWINE wc:BNE kmlp
5680 ADD R1,R6,#4:B knam3
5690.kblk2:EQUD 0
5700.kblk1:EQUB 0:ALIGN
5710.kstr:EQUS STRING$(30,CHR$0)
5720.kstr2
5730 EQUS"%W3,%DY %M3 %CE%YR.%24:%MI:%S
E ":EQUB 0
5740.ttab
5750 EQUB &FF:EQUS"     ASCII"
5760 EQUB &FE:EQUS"   Command"
5770 EQUB &FD:EQUS"      Data"
5780 EQUB &FC:EQUS"   PI Code"
5790 EQUB &FB:EQUS"     BASIC"
5800 EQUB &FA:EQUS"    Module"
5810 EQUB &F9:EQUS"    Sprite"
5820 EQUB &F8:EQUS" Abs. Code"
5830 EQUB &F7:EQUS"  BBC font"
5840 EQUB &F6:EQUS"Fancy font"
5850 EQUB &00:EQUS"          "
5860.untag:BL tagcl:B newd
5870.end ]:ENDPROC
```

RU

REVIEWS

Clares' Image Writer

Image Writer may well be the first word processor written specifically for the Archimedes. Mike Williams examines a pre-release version of this attractive software.

Perhaps the most essential application which any micro needs to provide is a decent word processor. Although both the old BBC micro favourites, View and Wordwise, are available for the Archimedes, there has, until now, been no word processor written specifically for this machine. All that is now set to change. Acorn will be supplying all registered Archimedes users with ArcWriter, but Clares look like being first in this race with their own Image Writer.

It is very important to make clear at the outset, that the version of Image Writer made available to us was a pre-release version, with some features only partially implemented (or not at all). Likewise the final documentation had still to be completed, and we were provided with some customised instructions to guide us on our way. Clares deserve some praise for their efforts to ensure we received a copy in time for a preliminary review in this issue.

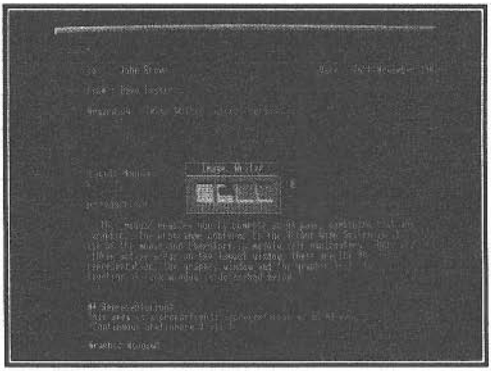
I have to say at the outset that Image Writer is VERY pretty, but I remain to be fully convinced of its real practicality as an everyday word processor. It uses colourful icon-filled menu windows for mouse control, but the icons are not always obvious when it comes to their meaning. Maybe simple lists of keywords would be more effective in pull-down (or pop-up) windows.

PAGE FORMAT

Initially (in our version) you are presented with the Page Format screen (the final version will take you straight into the Word Processor screen). The left of the screen shows the first page of your document at reduced size, while a table to the right shows current parameters for headers, footers and the like. Some parameters when selected just cycle through suitable values, others display a pointer on the reduced page for direct manipulation (e.g. header space). In practice I found it somewhat difficult to judge the final result of what I thought I was doing. Why not use the pointer to change a parameter (like header space) and see the effect change dynamically on the reduced image?

WORD PROCESSOR

At the bottom of the Page Format screen is a set of 9 icons to select other parts of Image Writer. The one I turned to first was the 'pen nib' image representing the Word Processor itself. The edit screen is an attractive 80 column display (white on dark blue) combining elements of both Wordwise and View. 'Rulers' can be inserted at any point in the text to position left and right margins, and (in the full release version) there will be a facility for setting tab stops for paragraph indents and tables. Maximum document width is determined by the 80 column screen width.



Word Processor Screen

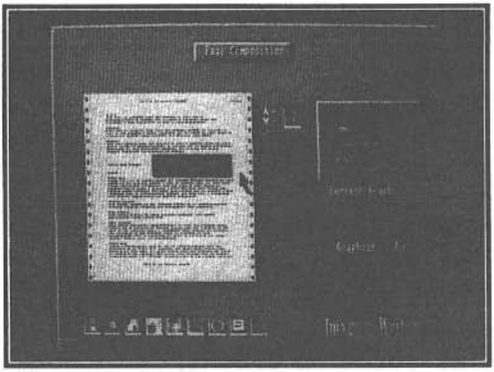
Disappointingly perhaps, text is not fully formatted (WYSIWYG style) on the screen, nor are any other effects like the use of bold or underline shown as such. Lines are correctly word-filled so that what you have is non-justified text. To see the true representation of your document you must select the preview function from a menu. Once there, all is displayed as specified (black on white) complete with bold, underlined text (or whatever else you have selected), and the mouse controls a VERY smooth scroll up and down each page.

Back in edit mode, all functions such as choice of justification and other styling effects are specified by means of embedded commands. These appear as blue background blocks in the relevant screen position, but there appears to be no way of distinguishing what 'highlight' marker has been specified in any position without pointing to an embedded command and selecting a menu to provide this information. Alternatively, you can view the text in preview mode as mentioned above. This arrangement does strike me as being less helpful to the user than one might wish.

Although the mouse can be used to move around any text you wish to edit, I suspect that most people would use the cursor keys for this purpose. They can be used, as you might expect, on their own, and in conjunction with Shift or Ctrl, to move up, down, left or right, a character or a word at a time, to the left or right margins, and to the start or end of your text.

Markers can be inserted to select a piece of text, and this can then be deleted, copied or moved. Marked text can also be copied or cut out for subsequent insertion in another document. It might have been nice to use the mouse to highlight sections of text, but this would apparently (according to Clares) have slowed down the operation of Image Writer too much.

Generally, anyone who is familiar with Wordwise will (with the exception of the rulers and the 80 column screen) find much that is familiar, even down to Ctrl-A to delete a character, and Ctrl-S to change case (though these might be implemented differently in the final version).



Page Composition Screen

SPRITES AND GRAPHICS

Where Image Writer certainly does appear to have more to offer than most word processors (on BBC micros at least) is the ability to define graphics screens to be incorporated into a page of text, and to use similarly pre-defined icons. Once into graphics mode, pressing the central (menu) button on the mouse generates a graphics menu very similar to that in Artisan (see last month's review of this graphics package from Clares). You can draw all manner of geometrical shapes, change colour and manipulate parts of the screen at will.

Clares say that Image Writer stores its graphics as VDU character streams to allow for subsequent re-scaling, impossible with the bit-image graphics used by Artisan. However, sprites are in bit-image form and these can be used as well. Given that the graphics facilities of Artisan and Image Writer do appear to be very similar, I would have thought a better option for the user might have been an enhanced and improved word processor, with the option, for those who wished to use it, to incorporate graphics generated by Artisan (or other means), onto text pages as required.

PAGE COMPOSITION

This screen allows picture frames to be positioned and sized on any text page, with any selected graphics image being positioned (and scaled) to fit. This works well, and there is also a special graphics preview mode which shows a reduced page complete with text and colour graphics. You can also use the normal preview mode for a full-screen detailed view of the page.

In practice there appeared to be annoying differences between the main and preview screens in the position and shape of a picture box, but Clares state that this is an insuperable problem that arises from the differing resolutions of screen and dot-matrix printer. Nevertheless, the differences are real, and I found them disconcerting. For example, a box scaled to 'landscape' proportions (i.e. horizontal) in the Page Composition screen appeared 'portrait' (vertical) when viewed on the preview screen.

CONCLUSIONS

I would stress again that my comments are based on a pre-release version of Image Writer. However, my overriding impression is that perhaps too much attention has been paid to the design and presentation of very attractive screen displays, at the expense of good, practical and comprehensive word processing facilities (though Wordwise enthusiasts will find the Word Processor option very familiar). I suggest you have a good look at your freebie ArcWriter before making your mind up, but if you want Image Writer's combination of text and graphics it is undoubtedly excellent value at just under £30. And if you get a chance to try it out, you'll probably be bowled over by its stunning use of graphics anyway.

Image Writer, £29.95 inc VAT
Supplied by:
Clares Micro Supplies
98 Middlewich Road,
Rudheath, Northwich,
Cheshire CW9 7DA.
Tel. (0606) 48511

ALL THE COLOURS OF THE RAINBOW

by Mike Williams

This month in our graphics spot, we'll take a look at another of the Archimedes' major graphics features, the wealth of colours available to the user. While long-suffering BBC micro owners have had to limit themselves to no more than six different colours in practice, plus black and white, the Archimedes provides a staggering 4096 different hues.

In the first instance, a major part of the change is from a digital to an analogue method of colour control. In a digital system with the three basic colours of red, green, blue, each of which may be either on or off, the possible combinations lead to the six colours red, green, blue, cyan (blue and green), magenta (red and blue), yellow (red and green) with black (no colours) and white (all three colours). With analogue control, the three basic colours may each be varied almost infinitely.

However, as always, nothing is quite as simple as it might first appear, and you will be rapidly disillusioned if you expect to have full use of 4096 colours in all modes. Part of the problem is that ARM Basic has tried to maintain complete compatibility with the screen modes and use of colour of the earlier BBC micros, while providing the best of Archimedes as well.

2 Colour Modes	4 Colour Modes	16 Colour Modes	256 Colour Modes
0	1	2	10
3	5	7	13
4	8	9	15
6	11	12	
18	19	14	
		16	
		17	
		20	

The numbers of colours available in each mode are shown in the table. The initial impression to be gained from this is that no more than 256 colours are available in any mode, though this is not correct. As a first step, we will deal this month with the two, four and sixteen colour modes as a group, and leave the 256 colour modes for a later article.

UP TO 16 COLOURS FOR TEXT AND GRAPHICS

Whichever mode you use in this group,choice of colour in the first instance is specified by a logical colour number, in the range 0 to 1 for two colour modes, 0 to 3 for four colour modes, and 0 to 15 for sixteen colour modes. Each logical colour number is associated with a particular screen colour. The default assignments are shown in the User Guide, but you can change the assignments so that any logical colour number may be associated with whatever colour you choose.

Logical colours are selected using the COLOUR statement for text, and the GCOL statement for graphics. In both cases adding 128 to the logical colour number chosen will specify a background shade, as opposed to a foreground colour. For example, specifying:

```
COLOUR 4
```

would select blue (by default) for all following text in a 16 colour mode, while:

```
COLOUR 131
```

would select abackground of yellow for all subsequent text. For graphics:

```
GCOL 1
```

would select the foreground graphics colour as red (by default), while:

```
GCOL 134
```

would select a graphics background colour of cyan. Again, these choices would remain in force for all future graphics (PLOT, DRAW, etc) until changed by a further GCOL command.

CHANGING COLOUR ASSIGNMENTS

There are, in principle, two ways in which the colours associated with the logical colour numbers can be changed, and in each case there are two quite different but equivalent ways of achieving the desired result. The first means of changing colour is compatible with earlier versions of BBC Basic and allows any of the 16 default colours (as listed in the User Guide for 16 colour modes) to be assigned to any logical colour number. This can be done with a VDU19 command. For example, to set logical colour 1 to be blue (colour 4) you would write:

```
VDU19,1,4|
```

Alternatively, on the Archimedes, exactly the same result can be obtained more elegantly by putting:

```
COLOUR 1,4
```

Note that the use of the keyword COLOUR in this context is quite different from its previous use to select a text or graphics colour for immediate use.

Although the simple colour assignment above is is both useful, and often necessary, it still limits us to a choice of just 16 colours. To be able to assign any of the 4096 shades of which the Archimedes is capable requires a different form of either the COLOUR or VDU19 statements. To create other colours you need to specify the proportions of red, green and blue to be mixed together. The format to follow is:

```
    COLOUR n,r,g,b
or: VDU19,n,16,r,g,b
```

where 'n' is the logical colour number to be assigned this colour, and 'r', 'g' and 'b' are the proportions of red, green and blue. Either instruction will do, but the COLOUR statement is simpler and shorter. However, the amounts of red, green and blue can only be specified in steps of 16 (0, 16, 32, etc), so it is often easier to think of this use of the COLOUR statement in the form:

```
        COLOUR n,16*r,16*g,16*b
```

where each of r, g, and b may vary from 0 to 15 in steps of 1. In this way we can assign any one of 4096 shades to any logical colour number. Remember, though, that either a simple 'COLOUR n' or 'GCOL n' will be needed to specify that the current text or graphics colour is 'n'.

COLOUR MIXING DEMO

The use of colour mixing in this way is demonstrated in the accompanying program, which also provides an easy way for you to experiment with the results of colour mixing. If you type in this program and run it, you will see one large white square on the screen, and below that small squares in red, green and blue respectively. Use the mouse to move the screen pointer to any of the three small coloured squares. Pressing the left hand button on the mouse will decrease the proportion of that colour, pressing the right hand button will increase that colour. The resulting mix of the selected proportions of red, green and blue is shown in the large square, and for reference the corresponding parameters for r, g and b are displayed at the bottom of the screen.

The part that handles all the colour changing is the procedure PROCmouse from line 360 onwards. This uses two nested CASE statements to increment or decrement the amount of each colour dependent upon the mouse button pressed and the colour being pointed to. Four COLOUR statements then re-assign the colours for the three small squares and the one large one. The program repeatedly calls this procedure to check the mouse and change colour as appropriate, and to re-display the boxes in the new shades. If you find that the colour changes are too fast for you, just increase the time limit (currently 5) at the end of line 180.

MORE COLOUR SELECTION

The VDU19 command, and in part the COLOUR statement as well, provide two further controls on the choice of colour. The command:

```
        VDU19,n,24,r,g,b
```

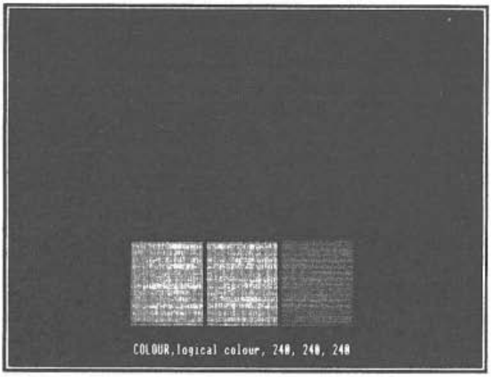
determines the mix of red, green and blue to use for a screen border. Such a statement is included at line 100 in the program, and produces an orange border. A further variation on this

```
        VDU19,n,25,r,g,b
```

determines the colour to be used for the mouse pointer on screen. It appears that, by default, logical colour '1' specifies the border colour of the pointer, and colour '2' the main pointer colour. In the program, these have both been re-assigned at lines 110 and 120 (to provide a pink pointer with a grey border). Alternatively, you can also specify the pointer colour with the equivalent:

```
        MOUSE COLOUR n,r,g,b
```

Changing pointer colour in this way means that you can define a solid colour pointer (edge and main colour the same), but remember that this will not then show up against a background of the same colour.



COLOUR,logical colour, 240, 240, 240

You should find it easy to use the program to mix alternative colours for both the screen border and the pointer, reading off the corresponding r, g and b values from the screen, and then amending lines 100, 110 or 120 as required.

Please remember that the information given above applies only to those modes which allow no more than 16 colours on the screen at a time. For 256 colour modes, things are quite different because of the conflict that arises from the desire to maintain compatibility with the older BBC micros, and the enhanced range of colours on the Archimedes. We'll take a look at that next month, but for now have fun experimenting with colour-mixing program.

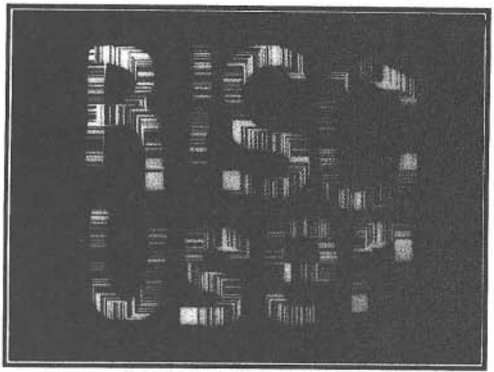
```
   10 REM Program Mixer
   20 REM Version B1.5
   30 REM Author  Mike Williams
   40 REM Based on an idea by Lee Calcraft
   50 REM Risc User December 1987
   60 REM Program subject to copyright
   70 :
   80 ON ERROR GOTO 530
   90 MODE 12:MOUSE RECTANGLE 0,0,1240,1
024:*POINTER
```

ALL THE COLOURS OF THE RAINBOW
ARCHIMEDES VISUALS

```
  100 VDU19,1,24,240,96,0
  110 VDU19,1,25,112,96,112
  120 VDU19,2,25,240,80,160
  130 OFF
  140 red=15:green=15:blue=15
  150 REPEAT
  160 PROCmouse
  170 PROCboxes
  180 TIME=0:REPEAT UNTIL TIME>5
  190 UNTIL FALSE
  200 END
  210 :
  220 DEFPROCboxes
  230 GCOL 1
  240 RECTANGLEFILL 345,420,590,500
  250 FOR colour=2 TO 4
  260 PROCrect(colour)
  270 NEXT colour
  280 ENDPROC
  290 :
  300 DEF PROCrect(c)
  310 GCOL c
  320 RECTANGLEFILL 200*c-55,200,190,200
  330 PRINTTAB(22,27)"COLOUR, logical col
our, ";16*red;",";16*green;",";16*blue;S
TRING$(8,CHR$32)
  340 ENDPROC
  350 :
  360 DEFPROCmouse
  370 MOUSE x,y,state
  380 p=16+(5-2*state)/3
  390 CASE state OF
  400 WHEN 1,4
  410 CASE POINT(x,y) OF
  420 WHEN 2:red=(red+p)MOD 16
  430 WHEN 3:green=(green+p)MOD 16
  440 WHEN 4:blue=(blue+p)MOD 16
  450 ENDCASE
  460 ENDCASE
  470 COLOUR 1,red*16,green*16,blue*16
  480 COLOUR 2,red*16,0,0
  490 COLOUR 3,0,green*16,0
  500 COLOUR 4,0,0,blue*16
  510 ENDPROC
  520 :
  530 MODE 12:REPORT:PRINT" at line ";ER
L:END
```

ARCHIMEDES VISUALS (continued from page 17)

RAINBOW SQUARES



Program RainbowSq

Graham Stanley's program (see listing 4) is extraordinarily compact for what it achieves. Essentially it is a mouse-driven painting program. But the brush paints with randomly coloured squares. Press the *select* button to draw, the *menu* button to increase the brush size, and *adjust* to decrease it. Buttons may be pressed simultaneously to create a variety of fascinating effects. To clear the screen, press *menu* and *adjust* together; and to quit, press all three buttons.

A305	SC 20	SP -	RMA -

Listing 4

```
   10 REM >RainbowSq5
   20 REM Rainbow Squares
   30 REM by Graham Stanley
   40 MODE 15:OFF:*POINTER
   50 VDU19,0,24,240,120,0:R=50
   60 REPEAT
   70 GCOL RND(64):MOUSE X,Y,Z
   80 IF (Z AND 1) AND R>0 THEN R-=.1
   90 IF (Z AND 2) AND R<200 THEN R+=.1
  100 IF Z=3 THEN CLS
  110 IF Z AND 4 RECTANGLE FILL
X,Y,R,R
  120 UNTIL Z=7:ON
```

SEND US YOUR VISUALS

We are running a competition for the best Archimedes visuals. Winners will receive a £20 retail token, and will have their programs featured in these columns. So send us your contributions without delay!

Send to: **Archimedes Visuals,**
RISC USER, Dolphin Place,
Holywell Hill,
St Albans, Herts, AL1 1EX

ARCHIMEDES INTO PC

by Ian Whiting

Ian Whiting, an experienced PC user, investigates Acorn's PC Emulator for the Archimedes. Can this turn the Archimedes into a competitive PC clone?

INTRODUCTION

Without any doubt the most prolific micro computer operating system for business users is Microsoft's MS-DOS, which is also sold by IBM as PC-DOS. MS-DOS requires an IBM PC (or XT etc), and uses the 8/16 bit Intel 80xx series microprocessor including the 8088, 8086, 80286 and 80386.

The IBM PC is a very successful design, and many other companies manufacture compatible models, giving access to the thousands of programs which are available for the PC. One of the best known PC clone manufacturers in the UK is Amstrad, with the 1512 and 1640 models. Acorn market their own PC clone for the BBC Master, the 512 Co-processor using the Digital Research DOS PLUS operating system, and this runs approximately 75% of MS-DOS programs.

EMULATION

It is the reduced instruction set of the RISC processor that gives the Archimedes its exceptional performance. It is ironic that this speed makes it possible to emulate complex instruction processors like the 6502 and 8088 in software.

The 6502 emulator is supplied with every Archimedes, allowing programs for the older BBC B to run in a 64K partition. Similarly, the PC Emulator duplicates the hardware environment of the IBM PC, including the 8088 processor, the BIOS and other support chips. Once the emulator has been loaded from disc (ADFS format), the Archimedes becomes an IBM PC clone until switched off or reset. MS-DOS can then be loaded from the second (IBM format) disc.

MEMORY

The Archimedes needs RAM for its own modules and the Emulator. The remainder is available for MS-DOS and the user's programs. With the Archimedes' default configuration and a minimum 1Mb of RAM (A310 not A305), the Emulator simulates a 360K PC. Unplugging some of the Archimedes modules, e.g. the sound scheduler, and changing the configuration as recommended in the instructions, will free a further 140K of memory.

The A400 series, with 1Mb or 4Mb of RAM, will provide what is the full standard quota of 640K for the PC. Today, few PC's (or compatibles) are sold with less than 512K.

DISC HANDLING

The Emulator formats 3.5" discs to the IBM 720K standard as used on the new IBM PS/2 computers. It will also read the more common 360K format providing you are able to attach a 40 track 5.25" disc drive to the Archimedes (Acorn advise against this unless the connection is fully buffered). The Archimedes disc controller is different to that of the IBM PC's. Consequently, some programs which directly access the disc controller, e.g. protected software, will not work.

For any serious development work, an Archimedes with just one disc drive drive can involve much disc swapping, but at least it offers as much disc space as a PC clone with two 360K drives. A Winchester hard disc can be segmented to contain both a MS-DOS partition and an ADFS partition. Acorn claims that the hard disc on an Archimedes operates at up to 12 times the speed of those on the standard IBM PC. Two utility programs, GETFILE and PUTFILE, are supplied, and these transfer files between ADFS and MS-DOS discs.

SOFTWARE COMPATIBILITY

Apart from the disc controller mentioned above, the PC Emulator offers excellent compatibility with the standard IBM PC. It ran, without any problem, all of the software I tried, including dBASE III (version 2.0), Foxbase (2.0), Wordstar (2.1), Lotus 123 (release 1), Quick Basic (version 3) and memory resident routines.

The CGA (Colour Graphics) emulation works well with the Archimedes colour monitor, but any programs expecting an EGA (Enhanced Graphics) or Hercules Graphic card will be disappointed. Programs requiring the full complement of 640K will require a 4Mb Archimedes.

PERFORMANCE

The original IBM PC does not enjoy a reputation for speed. Even so the PC Emulator is noticeably slower. This is confirmed by the benchmarks, see figure 1. Using Microsoft's GW Basic (supplied with the Acorn package), the Emulator performs at half the speed of the IBM PC for everything except plotting to a graphics screen (88%) and writing to the disc drive (100%).

	PC Emulator	IBM PC/XT	Perf. Index
Integer Maths	11.6	6.2	0.53
Real Maths	14.5	7.0	0.48
Trigs etc	111.3	52.5	0.47
Text display	187.0	89.8	0.48
Graphics	56.0	49.0	0.88
Store to disc	36.0	36.0	1.00

The performance index shows the speed relative to the IBM PC.

Figure 1. Performance Benchmarks
(times in seconds)

Interestingly, two 'performance indicator' programs, Norton SI and PC-TOOLS Info, which time a representative sample of 8088 instructions, report that the Emulator runs at 0.8 and 1.45 times the speed of an IBM PC respectively. It all depends upon the selected mix of instructions; multiplication instructions are faster, as are some BIOS interrupts. For comparison the Master 512 has speed indicators of 4.9 and 1.25 respectively (faster than the Archimedes).

DOCUMENTATION

The documentation supplied is limited to one meagre 11 page application note, and is inadequate for anyone with limited previous experience of MS-DOS. More specific information is needed on just how the Archimedes PC Emulator works. For instance it fails to mention GETFILE and PUTFILE. As the PC Emulator may well find a niche in schools whose experience is limited to BBC model B computers, more documentation should be supplied.

SUMMARY

The PC Emulator is an interesting product, and Acorn have much to be proud of both in the emulation program and in the versatility of the Archimedes in accommodating such an emulator. The compatibility is little less than excellent, but its pedestrian pace makes any serious word processing, program development, or large processing tasks unattractive.

The PC Emulator will appeal to you you need to run an MS-DOS program in order to understand the functions of that program rather than use it in anger. It will also find a place in schools which have standardised on the BBC range yet have a passing interest in MS-DOS. But for anyone who really needs an MS-DOS computer there are a number of clones available from £350 which are faster, 100% compatible, accept additional cards (e.g. Enhanced Graphics), and arrive with real manuals.

PRODUCT INFORMATION

The PC Emulator package contains two unprotected discs, the PC Emulator and MS-DOS including GW Basic, and some instructions. It is available now for £113.85 inc VAT. It needs at least 1Mb of RAM (i.e. an A310 or an upgraded A305).

Acorn also market an A310M which is a standard A310 with the PC Emulator and MS-DOS included. This retails at an additional £69 (inc. VAT) over the A310 price. Acorn had intended to release a 640K 80186 co-processor board for the Archimedes in 1988 but this product has been abandoned.

USING BASIC LIBRARIES

By Mike Williams

Organise your Basic programming by using libraries of procedures and functions.

One of the attractive features .implemented in Basic V on the Archimedes is the concept of function and procedure libraries. The idea is not a new one but is more usually found in association with compiled languages such as 'C' and Pascal. Libraries can have many advantages, both in the more effective use of one's programming time, and as a better way of organising the resources of the Archimedes.

A procedure or function library (and such libraries may contain both constructs) is simply a file containing a number of procedures and/or functions, usually linked by some theme or purpose (a graphics library, or a set of input/output routines for example). When writing a program, one simply has to ensure that any libraries are loaded ready for use, and any subsequent function or procedure calls will automatically be satisfied by reference to the loaded libraries, if the routines cannot be found in the main program.

HOW TO USE LIBRARIES

A library in Basic V is a saved Basic 'program' containing only functions and/or procedures (though REM statements may also be included for documentation). There are two ways in which libraries may be loaded, using the commands INSTALL or LIBRARY (see later for the differences between these commands). In both cases the command is followed by the file name of an appropriate library, for example (and note the essential quotes):

```
      INSTALL "Graphics"
or    LIBRARY "Output"
```

Furthermore, a string variable to which a suitable filename has already been assigned, may also be used, such as:

```
      Libname$="Graphics"
      LIBRARY Libname$
```

This opens up many possibilities - for example, library names could be specified in one or more DATA statements and read in for use as required.

Here is a simple example of a library in use:

Main Program

```
10 REM Library Demo
20 LIBRARY "procsl"
30 PROCone
40 PROCtwo
50 END
```

Two procedures, called PROCone and PROCtwo, are saved as a library called 'procs1'. The main program loads the library and calls each procedure in turn.

Library 'procsl'

```
 1 REM Library procsl
10 DEF PROCone
20 PRINT "one"
30 ENDPROC
40:
50 DEF PROCtwo
60 PRINT "two"
70 ENDPROC
```

THE TECHNICALITIES OF LIBRARIES

The INSTALL command will load a specified library into memory above the stack, and move the bottom of the stack (and HIMEM) correspondingly down in memory. As a result, INSTALL cannot be used if the stack contains relevant information - that is from within a function or procedure, or within any form of loop. In addition, INSTALLed libraries may only be deleted by quitting Basic (i.e. just type QUIT). It would seem best to reserve it for use in immediate mode, or even better in a BOOT or EXEC file. Alternatively, the LIBRARY command loads a library file into memory above the existing program, effectively using part of the variable storage area. LIBRARY can thus be used from within loops, functions and procedures, but libraries loaded in this way are deleted whenever CLEAR is executed, or when a program is RUN or CHAINED.

Clearly INSTALLed libraries are much more permanent. An INSTALLed library would be the obvious choice to support a suite of Basic programs that chain to and from each other. Since the LIBRARY command can be used from within procedures one could organise a library file of initial procedures, such that if any of these procedures were to be called, then it would dynamically load a further library file of supporting procedures. This would avoid unnecessary disc access at the start of a program, and would also use less memory.

Remember to avoid using the same variable names in different libraries (no problem if you consistently use LOCAL in all procedures for all truly local variables). Note, also, that the LVAR command will display the currently loaded libraries, as well as other information.

HINTS & TIPS HINTS & TIPS

Our regular column of hints and tips comes this month from Rob Barnes and Lee Calcraft. If you have come across any good tips while using the Archimedes, please drop us a line.

FITTING ECONET

Econet can be fitted to the Archimedes by using the readily available Master Econet upgrade. The ROM supplied with this upgrade is not needed since all necessary software is resident within Arthur.

SERIES ONE WIPE AND COPY

The effect of the *WIPE command on operating system 1.2 has been changed to give as default both V and C options (Verbose and Confirm), but not F and R (Force and Recursive). Use *HELP WIPE for more info. To turn off either of the C or V options, use the tilde (~) as follows:

```
      *WIPE filename ~C
```

This will cancel the Confirm option for this WIPE operation.

To alter its effect more permanently, use the new command ***SET Wipe$Options**. For example:

```
      *SET Wipe$Options ~C ~F ~R V
```

globally turns off the C, F and R options, leaving on only the V option. This setting is lost after a hard Break or power up.

The options for *COPY are defined in a similar way using ***SET Copy$Options**. Use *HELP COPY and *SHOW for further information.

LISTO 8

Typing the command:

```
      LISTO n
```

where n is an integer in the range 0 to 15 sets up a variety of options which take effect when Basic's LIST command is used. LISTO options greater than 7 list your program with line numbers stripped away. See the User Guide for further details. All lines containing GOTOs are (quite rightly?) flagged as errors.

SYS BY NAME OR NUMBER

Some of Arthur's most useful routines can be accessed from Basic using the SYS command, which in many respects is the equivalent of the assembler instruction SWI. Both forms of the call can be made either by number or by name. Thus:

```
      SYS 0,65
```

has the same effect as:

```
      SYS "OS_WriteC",65
```

They both call the routine *OS Write Character* to put character 65 ("A") on the screen. Note that in the second version, the call name must be given in quotes *exactly* as it appears in the Reference Manual. Any change of case will cause it not to be recognised.

R-KEY CONFIGURING

If you power-up your Archimedes while pressing the "R" key, then switch off and repeat the process, you will reset the machine to its factory defaults. See the article on "Configuring Archimedes" in this issue for further details.

BORDERING IN MODES 3 AND 6

If you create a screen border in modes 3 or 6 (e.g. VDU19,0,24,240,96,96 gives a red border), you will see that the border colour also appears in rulings between each line, so creating a third colour in these two-colour modes. The background between the rulings can be coloured as normal by using VDU19. For example VDU19,128,4,0,0,0 will give a blue background.

A FLUSHED MOUSE IS A HAPPY MOUSE

You may be familiar with the idea of a keyboard buffer, which holds sequences of keys pressed until a program has time to deal with them. Well, the mouse also has a buffer, as you may have discovered when trying to read in mouse data from within a program. To be sure of reading the right data, you can flush the mouse buffer just before it is read. To do this, use *FX21,9

POINTER TROUBLE

Typing *POINTER should put the pointer on the screen at any time. If it fails, and gives a **Bad command** message, it is likely that your Window Manager relocatable module has become software-unplugged. To restore it, type:

```
      *RMREINIT WindowManager
```

Then initialise using Ctrl-Break.

It looks from the User Guide (June '87), as if the call *FX106,1 can be used to turn the pointer on. In fact the command will only work if the pointer has been previously initialised with *POINTER. By the way, to simulate a defunct mouse, try *FX106,129 with the pointer already on the screen. You can restore it to health with:

```
      *FX106,1.
```

HINTS & TIPS HINTS & TIPS

RETURNING MODE

The new pseudo-variable MODE is used by Basic to return the currently selected mode. To find what mode you are in, type:

```
PRINT MODE
```

COLOURING TWIN

To give Twin's display a blue background, press function key f0, then hold down the Ctrl key, and press the following keys in sequence:

```
S@D@@@
```

This works by issuing a VDU19 colour command. Note that the "@" key also requires the use of Shift.

RESIDENT DEBUG

The Debugger (previously supplied in the Modules directory of the Welcome disc) is permanently resident on machines with the series one operating system. You can now enter the module at any time by typing:

```
*DEBUG
```

or you can call its various routines as star commands. The commands are documented in the Programmer's Reference Manual. But you can discover enough to use the module by typing:

```
*HELP COMMANDS
```

to give the command names, and then using these names with individual *HELP calls to discover the syntax. Here is a shortlist:

*MEMORY	A scrolling memory dump
*MEMORYA	Simple memory editor
*SHOWREGS	Displays ARM registers
*MEMORYI	ARM Disassembler

MORE FILE TYPES

Here are four extra file types to add to the list given in the first issue of RISC User:

&FEF	Diary	&FED	Palette
&FEE	Notepad	&FE0	Desktop

WHICH MACHINE ?

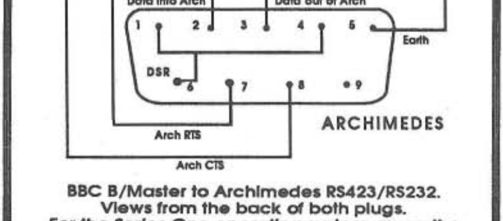
To distinguish an Archimedes from earlier versions of the BBC micro, use:

```
PRINT INKEY(-256)
```

This returns 160 with an Archimedes.

BEEB TO ARCHIMEDES TRANSFER

The accompanying diagram gives the wiring for Beeb to Archimedes transfers through the RS423/RS232 ports. Note that three pins of the Archimedes plug are wired together, but not to the lead itself.



BBC B/Master to Archimedes RS423/RS232.
Views from the back of both plugs.
For the Series One operating system, swap the connections to pins 6 and 8.

A simple transfer of individual Basic programs can be achieved as follows. Load your program into the Beeb as normal, then at the Beeb's keyboard, type:

```
*FX8,3
*FX3,1
LIST
```

Then, at the Archimedes, type:

```
*FX7,3
*FX2,1
```

and the transfer will take place. Then use Ctrl-Break then OLD to gain control of the Archimedes. Here we have used the slow speed of only 300 baud. You might like to try faster speeds (FX8,4 and FX7,4), but there may be corruption.

Note that a number of companies (including BEEBUG and Brainsoft) are advertising suitable leads, and software which will enable automated multi-file transfer at high speed.

L.G.C.

RISC USER magazine

MEMBERSHIP

RISC User is available only on subscription, with a special introductory rate until early 1988. Full subscribers to RISC User may also take out a reduced rate subscription to BEEBUG (the magazine for the BBC micro and Master series).

All subscriptions, including overseas, should be in pounds sterling. We will also accept payment by Connect, Access and Visa, and official UK orders are welcome.

Subscription Rates (12 Months)

	RISC User	BEEBUG (reduced rates)
UK, BFPO, Ch.Is	£12.50	£ 8.00
Rest of Europe &Eire	£18.00	£10.00
Middle East	£22.50	£11.00
Americas & Africa	£25.00	£11.50
Elsewhere	£27.00	£12.00

BACK ISSUES

We intend to maintain stocks of back issues New subscribers can therefore obtain earlier copies to provide a complete set from Vol.1 Issue 1. Back issues cost £1.20 each. You should also include postage as shown:

Destination	UK, BFPO, Ch.Is	Europe plus Eire	Elsewhere
First Issue	40p	75p	£2
Each subsequent issue	20p	45p	85p

MAGAZINE DISC

The programs from each issue of RISC User are available on a monthly 3.5" disc. This will be available to order, or you may take out a subscription to ensure that the disc arrives at the same time as the magazine. The first issue (with six programs and animated graphics demo) is at the special low price of £3.75. The disc for issue 2 contains all the programs from the magazine, including the complete menu utility, plus a music file and several demo graphics screens, all at the standard rate of £4.75.

The prices for magazine discs are shown below:

	UK	Overseas
Single issue discs	£ 4.75	£ 4.75
Six months subscription	£25.50	£30.00
Twelve months subscription	£50.00	£56.00

Disc subscriptions include postage, but you should add 50p per disc for individual orders.

All orders, subscriptions and other correspondence should be addressed to:

RISC User, Dolphin Place, Holywell Hill, St Albans, Herts AL1 1EX.
Telephone: St Albans (0727) 40303
(24hrs answerphone service for payment by Connect, Access or Visa card)